Dolog AKF125 → A120/A250
Type: AKF125EN
Version: 7.10
Configuration A250 (Vo. 1)
User Instruction

DOK–702086.35–1096

Translation of the German Description
DOK–700569.35–0196

Accompanying software package E-No. 424-275182

Documents in the software package

Kit 1

Documentation	Area of application
Installation User Instruction DOK-702082	Explains the usage and installation of the diskette s included.
How do you proceed? User Instruction DOK-702084	Serves as a ”red thread” through the documentation of the software packet and should be gone over before the start.

Kit 2

Documentation	Area of application
AKF125 for Beginners User Instruction DOK-702083	Serves to introduce new customers to AKF125. The user learns how to use the software in samll steps.
Short Form Guide A120 User Instruction DOK-702087	Tables for validity ranges and symstem markers, SFB-Formal operands for quick use onsite.
Sort Form Guide A250 User Instruction DOK-702088	Tables for validity ranges and symstem markers, SFB-Formal operands for quick use onsite.
Configuration A120 User Instruction DOK-702085	Contains the new features of the current version and explains th efunctions of th individual software menus for the configurer.

Kit 3

Documentation	Area of application
Configuration A250 (Vo1) User Instruction DOK-702086	Contains the new features of the current version and explains th efunctions of th individual software menus for the configurer.
Configuration A250 (Vo2) User Instruction DOK-707695	The explanation of the individual software menus will continued.
Masterindex User Instruction DOK-702089	Index of all documentation.

Notes

Application Note

Caution The relevant regulations must be observed for control applications involving safety requirements.
For reasons of safety and to ensure compliance with documented system data, repairs to components should be performed only by the manufacturer.

Training
Schneider Automation GmbH offers suitable training that provides further information concerning the system (see addresses).

Data, Illustrations, Alterations
Data and illustration are not binding. We reserve the right to alter our products in line with our policy of continuous product development. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us by using the form on the last page of this publication.

Addresses
The addresses are given at the end of this publication.

Copyright
All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying, processing or any information storage, without permission in writing by the Schneider Automation GmbH. You are not authorized to translate this document into any other language.

Trademarks
All terms used in this user manual to denote Schneider Automation GmbH products are trademarks of the Schneider Automation GmbH.

© 1996 Schneider Automation GmbH.

Terminology

Note This symbol emphasizes very important facts.

Caution This symbol refers to frequently appearing error sources.

Warning This symbol points to sources of danger that may cause financial and health damages or may have other aggravating consequences.

Expert This symbol is used when a more detailed information is given, which is intended exclusively for experts (special training required). Skipping this information does not interfere with understanding the publication and does not restrict standard application of the product.

Path This symbol identifies the use of paths in software menus.

Figures are given in the spelling corresponding to international practice and approved by SI (Système International d‘ Unités).
I.e. a space between the thousands and the usage of a decimal point
(e.g.: 12 345.67).

Abbreviatons

ABS	absolute Adressing
Adr.	Adresse (signal adresse)
AE	Block for one time actions
AZ	Block for cyclicel actions
AKF	Instructionlist, Contaktplan, Functionlist
ALD	Sequentiel Flow Chart with Diagnostics
ALS	Sequentiel Flow Chart
AWL	Instructionlist
AWP	User programm
BGT	Subrack
BSS	Serviceconnection for PC ore VS210
DAE	Diagnostics Block for one time actions (AE)
DAZ	Diagnostics Block for cyclicel actions (AZ)
DIB	Diagnostics Block for Block independent Diagnostics
DB0....9	SYM/KOM-Datablock for A120
DPB	Diagnostics Block for Programm Blocks (PB)
DSB	Diagnostics Structure Block
DW	Double word
I/O	Input– / Output signales (e.g. from a Module)
FB	Function block
FUP	Function list
FW	Floatingpoint
HW	Hardware (z.B. PLC)
IB	Initial value block
KB	Sequentiel block
KF	Sequentiel errorbit
KFW	Sequentiel errorword
KOP	Kontaktplan
KS	Kettenstatus für Simultankette
LZS	Run Time System
MW	Markerword
OB	Organisations Block
PB	Program Block
PaDT	Programming- and Diagnostics testequipment
RK	Controlloop
SFB	Standard-Function Block
SK	Step marker

SM	Systemmarker
SSP	Signal memory
SW	Software
SYM	symbolic Adressing
SYM/COM	Symbol und Comment
SZ	Step Counter
TB	Transitionsblock
TN	Teilnehmer
VBGT	Virtuelle Subrack (InterBus-S, Modnet 1/IS))
ZVT	Time organiasationtable (Controlle)
ZZ	Time Counter
<Return>	Applay the key Return
<Esc>	Applay the key Esc
<Ctrl>+<Alt>+<Del>	Applay in the same time the keys Ctrl, Alt und Del (beginning with Ctrl.and finnishing with Del)

Objectives

The functions of the software for configuring PLC stations are described. The documentation is set up like a reference document. Frequent consultation of the index is recommended.

Arrangement of this guide

Chapter 1 contains new features and a broad overview of the software structure.

Chapter 2 goes into how the software is used.

Chapter 3 contains actual functional descriptions of menu item **editing**.

Related Documents

A250
User Manual A250
804 BHB 000 00

A250
User Manual A250
Regeln mit Dolog AKF
804 BHB 001 00

A250
User Manual A250
Prozessperipherie Frontanschlusstechnik
899 BHB 000 00

A250
User Manual A250
Cable
899 BHB 001 00

A250
Blockbibliothek Standard Funktionblocks A250
Vol. 1 (AKF125 V4.x, ALD25 V4.1)
804 BSB 001 00

A250
Blockbibliothek Standard Funktionblocks A250
Vol. 2 (AKF125 V4.x, ALD25 V4.1)
804 BSB 002 00

Validity Note

These User Instructions apply to the AKF125 software, version 7.1, on the DOS operating system.

The current intention is for **remote control data** only to be edited with the **AKF125** configuration software and **not** with ALD25. Therefore, within systems U250 and UZ250, remote control modules **KOS140**, **KOS141** and **DEZ161** should **not** be used as REAL–TIME variants.
Correspondingly, the **KOS 20x** modules of the U120, Z120 and UZ120 systems are **not** to be used.

Table of Contents

Chapter 1 Introduction 1

1.1 General 2
1.1.1 Programming the A120 3
1.2 New features 4
1.3 Current information 5
1.3.1 General operating information 6
1.4 Software compatibility 9
1.5 Scope of the first menu level 10

Chapter 2 Operation 15

2.1 Using the mouse 16
2.2 Using the keyboard 17
2.2.1 US/German keyboard 17
2.2.2 Keyboard sections 17
2.2.3 Line editor, ”Line editor” 19
2.2.4 Screen copy, ”Screensave” 20
2.2.5 Key input memory, ”Learning” 21
2.2.6 Key text buffer, ”Key macros” 22
2.2.7 Auto Repeat Function 22
2.2.8 Special keys 23

Chapter 3 Programming 25

3.1 Introduction 26
3.2 Dolog AKF Main menu 27
3.2.1 Pulldown menus 28
3.2.2 Help function 28
3.3 Edit 29
3.3.1 Overview 30
3.3.2 Blocks 33
3.3.3 Symbols and Comments 154

3.3.3.1 Search Function 160
3.3.3.2 Search Operand 161
3.3.3.3 Erase Field / Line 161
3.3.3.4 Copy Area 162
3.3.3.5 Erase Area 163
3.3.3.6 Move Area 164
3.3.3.7 Copy modified 165
3.3.3.8 Search Text 167
3.3.3.9 Replace Text 168
3.3.3.10 Search/Replace Again 168
3.3.3.11 Insert comment line 169
3.3.3.12 Generate Symbols 169
3.3.4 Equipment list 172
3.3.4.1 General information 175
3.3.4.2 Possible modules and their parameters 178
3.3.4.3 Equipment List editing functions 194

Index 229

Chapter 1
Introduction

This chapter contains general information about this manual. It also gives the general structure of the different functions of this software.

1.1 General

This programming guide is intended as reference for working with the software Dolog AKF → A120/A250.

This software consists of two parts:

- Dolog AKF → A120
- Dolog AKF → A250

A description of the Dolog AKF software is given below → A250 for A250.

The software is used for structured programming of PLC user programs with the aid of up–to–date window technology (with pulldown menus).

☞ **Note** After installation you can press the <F10> key twice for an overview of Help texts.

☞ **Note** In this overview, look under ”Information on Help” and ”Dolog AKF software overview” to obtain general information about Dolog AKF software and the operator interface.

☞ **Note** In addition you can call up information on differences between the versions, even for the previous version, via an item called ”Differences between Versions”.

The texts in the ”Programming” chapter are organised according to the structure of the Dolog AKF main menu, from left to right. Within the description, the alphabetic index explains where to find function descriptions in the text.

1.1.1 Programming the A120

AKF125 can also be used to program automation system A120 (if using an ALU 204 / ALU205).

Furthermore, from AKF125 version 6.0 the **AKF12** (version 6) software package for configuration of an A120 (i.e. with ALU 200 / 201 / 202 etc.) is integrated.

Method of operation
For reasons of increased system unifomity and improved processor performance in the A250 subsystem A120, the module ALU 204 / ALU205 was created. The ALU 204 / ALU 205 holds the ALU basic software for the A250 system and the A120 system (subracks, I/O components, etc.) is installed in the hardware. It thus combines the wide range of features of the A250 system software with the compact hardware of the A120 system.

Configuration (ALU 204 / 205)
The ALU 204 / ALU 205 is configured with the aid of ALD25 as an additional ALU in the A250 system. Configuration takes the same form as with the ALU 151 ... 154. It is merely that the available modules and expert SFBs are different. If the ALU 204 / ALU 205 is chosen as the central processing unit, the modules of the A120 system appear in the Equipment List Editor of the A250 system. A detailed description of this editor can be found in Chapter NO TAG

Using decouple module DEA 201, it is possible to operate an A120 secondary backplane DTA 20x on an A250. The communication for this is via Modnet 1/SFB.

All the SFBs of the A250 system are available for use with the ALU 204/ ALU 205. The only exceptions are the expert SFBs. These are available in modified form. A precise description of the SFBs concerned can be found in the block library.

Configuration (AKF12)
Switch to software package AKF12 via the new menu command ”ALU Group” in the SETUP / PLC-STATION menu window.
Switch back to the software package AKF125 for A250 via the menu command ”ALU type” in the SETUP / PLC STATION menu window.

1.2 New features

AKF125 version 7.1 compared to version 6.0
If you have received a new version of the configuration software and installed it on your PaDT, stations that have already been configured can be imported at any time via the Export and Import function. Since Version 4 a station can be imported by the new configuration software by making an entry in SETUP. Following a security inquiry, the station is then irreversibly converted to the new version.

- System features
  - Improved flash drive initialization
  - Improved loading time (loading blocks that are not present)
  - Swapping of IEC blocks (MMSE) to 2nd MB
  - PLC memory automatically standardized before bootload
  - Master/slave function for Profibus added
  - Miscellaneous error corrections

- Configuration features
  - Real time status display
  - Improved error localization during PLC diagnostics (InterBus)
  - Stream DBs imported into the 2nd MByte (UZ250 and UZ251)
  - Loading of Expert data per group (up to 8 KOS modules)
  - Pseudo–module DXX for entering third–party modules into the equipment list
- New blocks:
  - SFB465 (BKF1) modified for operation with BKF 102.

- Interface for modules:
  - Interface to the BKF102 interface module for Modnet 1/IS (InterBus; installation remote bus and A120 to remote bus)
  - Interface to the NOK 116 module for cam controller (data structure).

1.3 Current information

You can find additional current information on **differences between versions** and **system limits** in the Help texts by entering the following keystrokes:

<F10>, <F10>, <PgDn> (”differences between versions”)

All data relevant to the user program, such as:

- Equipment List
- Data Structures
- Symbols, comments and initial values
- Blocks (OBs, PBs, FBs)
- Standard function block - LIB
- I/O–card definition list

can be **imported** and **ex**ported as ASCII files.

The standard method for transferring **old** stations (AKF125/ALD25) to the current version is to use the **Import and Export** function.

1.3.1 General operating information

- Oerating **more than one** PaDT on the PLC (RS232 / Modnet 1/SFB) simultaneously is **not** allowed.
- The PLC remains in boot mode after bootloading. Action: Remove the battery for a few minutes.
- Exchange Online of control technology OBs is only possible when the application is disabled (CRT parameter on SFBs O_REG, O_TIME, O_INTR must =0).
- The contents of system marker SMD126-131 are false on the ALU154/204/205 when Boot EPROM Index =.01.
- A250 slot addresses must be specified greater than 16. If this is not done, incorrect action occurs when downloading via Modnet 1/SFB.
- SW time monitoring must be allowed to remain active. If not, various kinds of incorrect actions can occur with individual ALUs if program runtime exceeds the specified watchdog time (EQU. LIST/<Ctrl.+Return>/Parametrize PLC).
  - Possible incorrect actions include:
    -ALU154 goes to boot mode
    -The green light on the ALU151/152/153 goes off; the peripherals are shut down; but the user program keeps running.
    -The green light on the ALU204/205 goes off until next time the watchdog is triggered and then comes on again; the peripherals are shut down until next time the watchdog is triggered and then switched on again; the user program keeps running.
  - Action: Bootload and load.

- Whilst using the online functions of PLC No. 1, switching over to some other PLC No. 2 is **not** allowed.
- When a program is transferred from AKF125 V4.0 or older, all indirect block calls of the SFBs IN, OUT, ANAIN, ANAOUT must be reprocessed (either by re–entering them or using Export and Import).
- General advice on importing AKF Stations created with version 3.04 or older **without** TXT102 and **without** POS102/112):
  - Export all blocks, data structures and equipment list under version 3.04 or older.
  - Erase the STATION.DBF file from the station directory.
  - Import all data structures, the equipment list and the blocks (in that order) under new configuration software.

- General advice on importing AKF Stations created with version 3.04 or older **with** TXT102:
  - Export all blocks, data structures and equipment list under version 3.04 or older.
  - Change the entry ”TXT102” into ”TXT1x2” in the Equipment List export file.
  - Change parameter TN (on reach thru) from ”TXT102” to ”TXT_KOPP”.
  - Erase the STATION.DBF file from the station directory.
  - Import all data structures, the equipment list and the blocks (in that order) under new configuration software.

- General advice on importing AKF Stations created with version 3.04 or older **with** POS102/112:
  - Export all blocks, data structures and equipment list under version 3.04 or older.
  - Delete the structure definitions for data structures POA and MPPA.
  - Erase the STATION.DBF file from the station directory.
  - Import all data structures, the equipment list and the blocks (in that order) under new configuration software.

- If on changing to new software with more features (more memory utilization for firmware) there is insufficient memory available, additional memory may possibly be freed by reducing **signal memory** and adapting **task data memory** to system requirements (EQU.LIST/<Ctrl.+Return>/Parameterize Central Controller or ...../ Parametrize PLC).
  - If task data memory is too **small**, the processing of time-controlled or interrupt–controlled OBs does **not** start.
  - If watchdog time is too **small** as far as time–controlled OBs are concerned, the user program **halts**.
- The use of DOS feature Doublespace is **not** recommended.

- Compatibility with existing products

  - Exchange Online is only possible with the same version (between PaDT and PLC).
  - Online writing functions (Set, Force, Triggered Recording, Process PLC / Load program signals) are disabled if you have version 3.x in the PLC and version 4.x, 6.0, or 7.x on the PUTE.
  - Status display (Dynamic Status, Online–List) is terminated if pointer references have to be displayed and you have version 3.x in the PLC and version 4.x, 6.0, or 7.x on the PUTE.

- Restrictions in the case of Near-DBs with version 7.x on the PaDT:
  no PROCESS PLC DATA with V3.x and V4.01 in the PLC
  no READ OUT PLC without Link and Load
  with V3.x, V4.01, V4.2, V4.21, V6.0 in the PLC.

1.4 Software compatibility

AKF125:

The software AKF125 is compatible with the packages shown below without restriction.

- COMAKF with effect from V4.5 (applies to AKF with effect from version 4.0)
- DEA-FW with effect from Index 10 when using TXT

In respect of migration from old to new software see also Chapter 1.3.1

1.5 Scope of the first menu level

AKF125 (A250)

The first menu level has the following contents:

(Contained in this volume)

Edit

- Overview
- Blocks
- Symbols and Comments
- Equipment List
- Title Block
- Replacement of Signals
- Compile (blocks)
- Data structures
- Run–time system

(Contained in Vol.2)

Load

- Link Program
- Program to PLC
- Exchange online
- Read equipment list
- Read Out PLC
- Process PLC data
- Compare
- Set date / time
- Process EPROM/FLASH
- Bootload
- Expert Data

Online

- Start PLC
- Stop PLC
- Dynamic status display
- Online list
- PLC status
- PLC diagnostics
- Communication operating functions
- Run–time system

Print

- Overview
- Program protocol
- Symbols and Comments
- Cross–reference list
- Signal assignment list
- Equipment list
- Complete documentation
- PLC data
- Data structures

Special

- Table of contents
- Backup the station
- Restore a station
- Erase a Station
- Format diskettes
- Copy diskettes
- Erase Files
- Copy files
- Import
- Export
- Further process PLC data
- Duplicate station
- operating system MS-DOS
- System information
- End station handling

Setup

- System
- PLC station
- PADT station
- Networking
- Print
- Colors

AKF125 (A120)

Edit

- Overview
- Blocks
- Symbols and Comments
- Equipment list
- Title Block
- Replacement of Signals
- Compile (blocks)
- Compile data blocks

Load

- Link Program
- Program to PLC
- Exchange Online
- Read Equipment List
- Read–out PLC
- Compare
- Set date/time
- Process EPROM
- Bootload

Online

- Start PLC
- Stop PLC
- Dynamic state display
- Online–List
- PLC status

Print

- Overview
- Program protocol
- Symbols and Comments
- Cross–reference list
- Signal assignment list
- Equipment list
- Complete documentation

Special

- Table of contents
- **Backup** the station
- **Restore** a station
- Erase a Station
- Format diskettes
- Copy diskettes
- Erase Files
- Copy files
- **MS-DOS**
- System information
- End station handling

SeTup

- Plant
- PLC station
- Print
- Colors

Chapter 2
Operation

Dolog AKF software is used for programming on standard IBM-compatible PCs under the MS–DOS operating system.
The work is performed with a mouse and/or keyboard.

2.1 Using the mouse

In addition to the typewriter–style keyboard, line editor and arrow keys, you can also use the mouse.
Please note that a parallel mouse (”bus mouse”) needs its own slot in the PaDT and has to be configured.

If using a P810 as a PaDT, you need a Microsoft PS/2 mouse.

Mouse operating functions
The **left** key is used to call and confirm a function. It is therefore equivalent to the <**Return**> key.
The **right** key is used to cancel or break, and is equivalent to the <**Esc**> key.

☞ **Note** The mouse should be connected to COM2 as standard. COM1 is required for PLC bootloading at least.

Warning If the mouse is operated on COM1 (only meaningful when programs are called with ...../NOSPS), the key combination <Ctrl>+<Return> (open menu) is not available.

2.2 Using the keyboard

2.2.1 US/German keyboard

The following key legends are different on the German and English keyboards:

US / UK keyboard	German keyboard
<Esc>	<Eing Lösch>
<Ctrl>	<Strg>
<Home>	<Pos1>
<End>	<Ende>
<Prtsc>	<Druck>
<PgUp>	<Bild ↑ >
<PgDn>	<Bild ↓ >
<Ins>	<Einfg>
<Del>	<Lösch> or <Entf>

2.2.2 Keyboard sections

The programming panel keyboard is divided into three different sections:

❒ **Typewriter keyboard and control keys**

The typewriter keyboard is located in the middle (light keys) and is surrounded by darker control keys. Only the white keys cause a display of the imprinted symbol to appear on the screen when they are pressed.
With the Dolog ALD Software, these are used to operate the line editor, the key macros and the reference characters (see ”Special Keys” Chapter 2.2.8) and to enter text.

❒ **Function keys**

These play a lesser role in pulldown menu technology, since all functions are selected with the mouse, reference characters or arrow keys.

F10 is used to call the context dependent help texts.
These keys are located above or alongside the typewriter keyboard (depending on the PaDT model) and are labeled <F1> ... <Fx>.

❒ Numerical pad

The arrow keys and numerical pad are located to the right of the typewriter keyboard. You can toggle between the numerical pad and the cursor pad with the <NumLock> key.
Of those in the numerical pad, the arrow keys are needed chiefly for the Dolog ALD software.
They are used among other things for selecting the various menu lines (see below).
The numbers are used to call Graphics characters (e.g. in the title block): <Alt> + <number>.
The <ALT>+number combination can also be used for defining keystroke sequences.(Chapter 2.2.5)

2.2.3 Line editor, ”Line editor”

The line editor helps you to input text and/or numbers and consists of combining the use of control keys and the typewriter keyboard.

It places the following editor functions at your disposal:

<Ctrl>+<A>, <Home>	Cursor to the beginning of the input line
<Ctrl>+<F>, <End>	Cursor to the end of the input line
<Ctrl>+<D>, < → >	Cursor to the right
<Ctrl>+<S>, < ← >	Cursor to the left
<Ctrl>+<M>, <Return>	End of input
<backspace>, < ⇐ >	Delete character to the left
<Del>	Delete character at cursor
<Ctrl>+<Y>	Delete input
<Ctrl>+<R>	Restore original text
<Ins>	Toggle between insert/overwrite

If the first character during input is a normal key (ASCII character), the input field is automatically deleted. The above–mentioned <Ctrl> keys leave the contents of the input field unchanged.

Using the <Esc> key interrupts input and the previous text is restored.

You can use the key macros (see 2.2.6) to insert stored text (e.g. symbols, comments, hardware addresses, file names, in part or in full).

2.2.4 Screen copy, ”Screensave”

This function is used to copy the complete current screen contents into a file. No printer need be connected. The file is stored in the directory of the current station in IBM-ASCII format. The copy (file) thus generated can then be edited using a text editor.

Step Simultaneously press the <Alt>+<F2> keys to generate a screen copy.

The first screen is saved under the name ”BILD0.BLD”. Screen 2 is called ”BILD1.BLD” etc.

☞ **Note** If you leave Dolog ALD and start over, screen numbering begins again at ”BILD0.BLD”. This means that copies created earlier with the same name will be **overwritten**.

2.2.5 Key input memory, ”Learning”

This function can combine up to 40 procedures (key inputs).

Step 1 Simultaneously press the <Alt>+<F8> keys to activate the function.

Step 2 Press the key combination to which input sequences are to be assigned. You can assign <Alt>+<0>, <Alt>+<1> etc. up to <Alt>+<9>.

Response. ”Learning” appears on the status line.

Step 3 Choose the required menus with the reference characters (up to 40 entries per assigned key).

Step 4 Press <Alt>+<F8> to end learning.

Response ”Learning” disappears from the status line.

Step 5 Press <Alt>+<0> or <Alt>+<1> etc. up to <Alt>+<9> to run the procedures you have saved.

☞ **Note** The menus should be selected with reference characters in order to make the learned functions independent of the current menu bar position.

2.2.6 Key text buffer, ”Key macros”

This simplifies the editing of symbols, comments, hardware addresses etc. Frequently used absolute or symbolic addresses can be stored as text. Proceed as follows.

Step 1 Press <Alt>+<F9> to activate the function.

Step 2 Press the key combination to which text is to be assigned. You can assign <Alt>+<A>, <Alt>+<B> etc. up to <Alt>+<Z>.

Step 3 Enter up to 19 characters of text.

Step 4 Press <Return> to save.

Step 5 Press <Alt>+<A>, <Alt>+<B> etc. up to <Alt>+<Z> to call the stored text to the cursor position.

At step 4 you can break off input by pressing <Esc>.

Example: the text **Valve** was stored. It was inserted as part of symbol/comment inputs (**Valve** 1 on, **Valve** 2 off, **Valve** 1 half, ...).

2.2.7 Auto Repeat Function

Most of the keys on a PaDT are equipped with the auto repeat function:
When you press down a key the symbol on the key itself is displayed on the screen or a corresponding function is executed. If the key is held down for longer than about 0.5 s, the key function is repeated up to 10 times a second until the key is released.
Therefore take care not to use this key function unintentionally.

2.2.8 Special keys

The following keys are of particular importance for controlling menus:

- **Reference characters**
  Reference characters are used to select and execute a menu directly from the typewriter keyboard. You access the corresponding function by inputting the specially marked capital letter.The highlighting colors may be adjusted in the ”SeTup” and ”Colors” sections.

- **<Ctrl>+Reference characters**
  Within an editor, the individual functions can be called with <Ctrl>+reference character, without having to open the menu.

- **Arrow keys**

  With the arrow keys (movement keys, cursor keys, direction keys) < ↑ >, < ↓ >, <→>, <←> you can select individual menu lines.
  The keywords are selected using the latter in the helptext.
  (When editing) A key depression corresponds to:

  < ↓ > one line down (in the same column)
  < ↑ > one line up (in the same column)
  <→> one column to the right (on the same line)
  <←> one column to the left (on the same line)

  Make sure you have switched the <NumLock> key off, so that the keys respond. When editing a block you can move the cursor with the mouse as much as you like without restriction.

- **<Return> key, <Ctrl>+<Return>**
  The <Return> key (also <Enter>, <Cr>) is used to call certain functions or to accept (translate). You can also ”toggle” with it (see below).
  Call
  - within the help function in order to call the help texts for the selected keyword
  - in the pulldown menus in order to call the selected menu line (menu function)
  - in ”Edit” (blocks, equipment list etc.): <Ctrl>+<Return> calls the editing function windows.

Exit, Accept

- when editing a network and when making entries with the line editor, press the <Return> key to accept the parameter, hardware address, comment, etc. you have just entered.

❒ **”Toggle”**

Many menu functions provide direct selection of prepared settings. Thus, for example, you can switch between ”off” and ”on” or ”IL”, ”LD”, ”FBD” directly (”toggling”), without having to enter text.
You can toggle in the following manner:

1 Type the specified reference character or
2 Go to the line using the arrow keys and toggle by pressing <Return>

The <Return> key is equivalent to the left button on the mouse.

❒ **<Esc> key**

Using the <Esc> key you can interrupt, i.e. break off, any function or input within the Dolog ALD software.
The <Esc> key is equivalent to the right mouse button

⚠ **Caution If you are at operating system level, you can interrupt only with <Ctrl>+<C>.**

❒ **<Ins> key**

In the line editor, in the case of comments and headers, you can use this key to toggle between the ”Insert” and ”Overwrite” text input modes.
In the block editor, ”Insert” has the effect of automatically performing a vertical spread in the network.

Table of Important Keys

At this stage it is not appropriate to list the most important special keys and key combinations. The same key combinations can perform different functions. When questions arise, please consult the index, where a key assignment table can be found. In addition, the explanation of each menu describes the available keys and their meanings.

Chapter 3
Programming

This chapter describes programming with the software DOLOG AKF for the A250.
The chapter is intended as a reference for the person configuring the system. Its structure follows the menu structure. Menu items that appear more than once are only discussed the first time they occur. The index contains the page numbers for descriptions of menu items that occur more than once.

3.1 Introduction

The individual menu points are described in the following order:

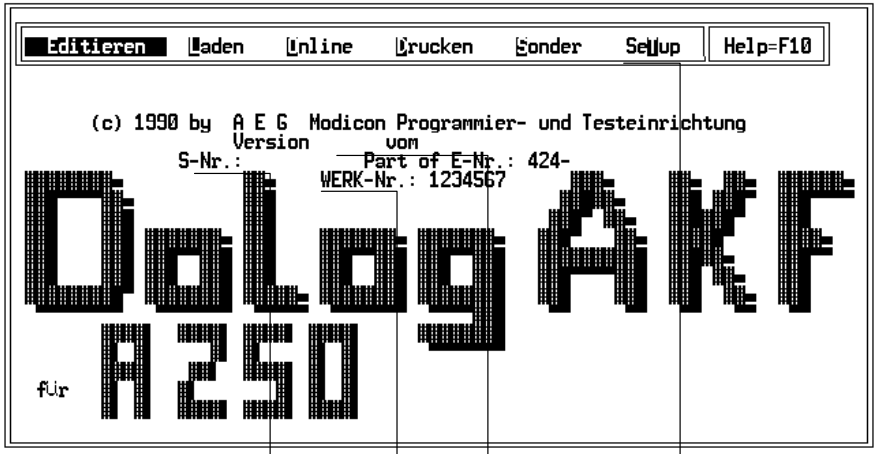
❒ Dolog ALD main menu	Chapter 3.2 (Configuration Vol. 1)
❒ Edit	Chapter 3.3 (Configuration Vol. 1)
❒ Load	Chapter 3.4 (Configuration Vol. 2)
❒ Online	Chapter 3.5 (Configuration Vol. 2)
❒ Print	Chapter 3.6 (Configuration Vol. 2)
❒ Special	Chapter 3.7 (Configuration Vol. 2)
❒ SeTup	Chapter 3.8 (Configuration Vol. 2)

First you will be shown how to select the menu line required (the ”path” from the main menu). The functions which you can execute at this point (programming, documentation, backup etc.) are then described.

This symbol shows how you can select the described functions. The steps are always given starting from the main menu.
e.g.:
- ”Edit”, ”Equipment List”, ”Controller”, ”Marker Bits”
means:
Select the ”Edit” menu,
Press Return to confirm the ”Equipment list” line,
Select ”Controller” and confirm by pressing Return,
”Marker Bits” is the function which is now described.

Note Some characters are written as capitals in the following description. They denote reference characters:
e.g. SeTup

3.2 Dolog AKF Main menu



Status line, in this case the path of the current system/ station; depending whether installed on
A:
B:
C:
D: or E:

Ser. No. of the diskettes as shown on slipcase

Ser. No. of your softwarepakkages (in accordance with software license agreement)

Version No. of the present software

Pulldown menus can be selected with reference characters, arrow keys + Return, or a mouse

3.2.1 Pulldown menus

Individual menu lines can be selected and called with reference characters, Arrow keys and <Return>, or the mouse (refer to chapter called ”Operation”). Menu lines that cannot be called at this time are marked in the ALS editor with the character ≡ (e.g. on–line functions when PADT is off–line).

3.2.2 Help function

Help texts describe a function. They tell you what can be carried out with the currently selected menu line or the current window. Help texts are also available for the higher menus.

The key <F10> is always used to display the help text belonging to a menu line selected from the pulldown menu.
This text may contain things known as keywords.
These are terms and expressions that have a further explanatory help text attached to them. Use the arrow keys to select them and press <Return> to call them. This makes the help texts easy to follow. For the most part, these texts are reminders for Dolog ALD / AKF beginners. Keywords are highlighted in color.
You can access the previously selected help text with <Alt>+<F10>.
Press <F10> again, which brings you to ”Overview Help Texts”. Ths shows you the keywords you can select, without menu lines of their own.
You may call references for any keyword in the index (index for help texts).

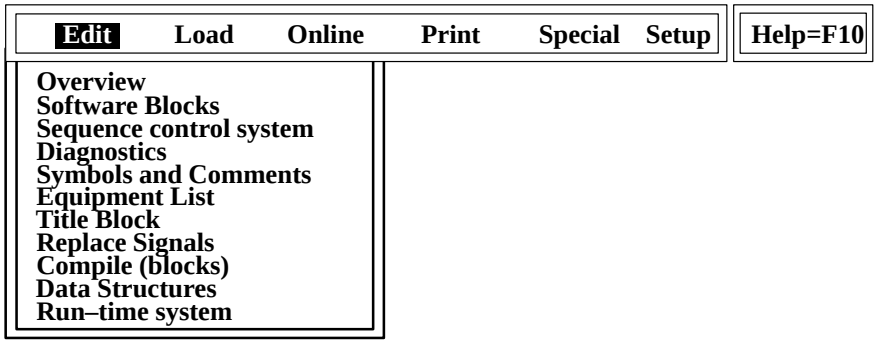
The bottom edge of every help window contains a line in which the available functions are listed:

<↓>, <↑>, <←>, <→>	Select keyword
<Return>	Call help text for the keyword
<Esc>	Terminate and exit the help window
<F10>	Overview Help Texts
<Alt>+<F10>	Call previous help text
<PgUp>, <PgDn>	Scroll backward and forward in the help text

3.3 Edit

The editing functions in pulldown menu ”Edit” are used to create your user programs (AKF editor), your sequences for the sequence control system (SFC editor) and the diagnostic elements (diagnostic editor), including the station documentation.

You can select and call up the following menu lines:



Caution The station specified under ”SeTup”, ”PLC station” is edited. The equipment list must be edited before the blocks.

3.3.1 Overview

- ”Edit”, ”Overview”

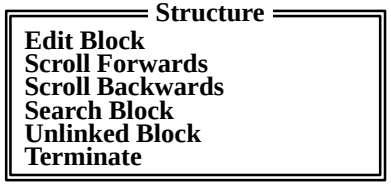
After selecting this function, a graphic program overview is created. In the case of addressing type SYM, the resulting display is symbolic. This overview makes clear how the complete program is structured, i.e. where and how often each block is called by other blocks. This function provides for shifting the image, selecting with the cursor (with the possibility of branching to the editing mode) and block search.

The overview is generated from OB1. The display is sorted by networks. If several blocks from the same network are called, they may not necessarily be displayed in the correct order.

Program structure

- ”Edit”, ”Overview”, <Return>

a) After pressing <Return> or <Ctrl>+<Return>, a menu with the following functions appears:



Functions are called with <Return>, the reference characters or the left mouse button.

b) You can bypass the menu and call menu items directly with <Ctrl>+<Reference character>, e.g. <Ctrl>+<**S**> for ”**S**earch Block”.

The following keys also have an effect:

<Esc>	Terminate
<Arrow keys>	Move cursor
<PgUp>	Previous page
<PgDn>	Next page

Edit Block

- ”Edit”, ”Overview”, <Return>, ”Edit Block”

You can use this function to edit a (subordinate) block which you have just selected (e.g. with <PgUp>/<PgDn>). Press <Ctrl>+<Return> to access the Block Editor.

Caution The logical functions provided are: ”And”, ”Or”, ”Exclusive–Or”, FBD elements, conditional and unconditional PBs, SFBs and user–defined FBs (conditional/unconditional). SFBs are system specific AEG blocks, that are called using either their SFB identifier (e.g. SFB1) or their SFB names (e.g. VIP+). (A list of SFBs can be found using help text (F10, F10, STDFB library). The sequence editor (SKB/SFB375) should not be called.

”Terminate” takes you back to the overview. The overview has then been updated.

Scroll forward / backward

- ”Edit”, ”Overview”, <Return>, ”Scroll forward”
- ”Edit”, ”Overview”, <Return>, ”Scroll backward”

Select either of these functions to scroll the screen up or down.

Search block

- ”Edit”, ”Overview, <Return>, ”Search Block”

Select the block you want with this function. For multiple calls the first call will be given. With <+> in front you can search forward and with <-> in front you can search backward.

Unlinked blocks

- ”Edit”, ”Overview”, <Return>, ”Unlinked blocks”

This function is used to call a selection window which contains the blocks that are not linked and therefore not shown in the overview. A block can be called with the arrow keys and <Return>. This automatically causes the block editor to branch.

Note This function does not apply to AZs and AEs.

Terminate

- ”Edit”, ”Overview”, <Return>, ”Terminate”

This function ends the overview and takes you back to the Edit menu.

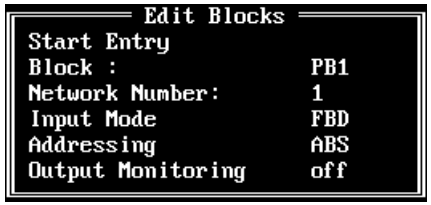
You can also exit the overview with <Esc>.

3.3.2 Blocks

”Edit”, ”Blocks”

This pulldown menu is used to generate the actual program. You can use the editors to program individual blocks in the display format you have selected. A description of the block editors follows, after which the individual menu lines are explained.

You can select the following functions in this menu:



Select the menu line ”Block” and confirm with <Return>. You can then:

- use the line editor to enter the block you are going to edit and confirm with <Return> (e.g. OB1, PB55, FB368, AZ3, AE6, KB8). Standard function blocks and transition blocks are not selectable

☞ **Note** If the block you have set up is not present, it will be created (i.e. when you select ”Start Entry” and <Return>, the message ”Block End” or ”BE” is displayed). In this case you are given a choice of editing functions if you press <Ctrl>+<Return> again.

- obtain a selection window containing all the blocks available in the station, by entering a space and <Return> at ”Block”. Then use the arrow keys and <Return> to choose the block you want to edit.

Block editors

The available block editors are the IL, LD and FBD editors (depending on input mode).

The block types OB, PB and FB can be input and reproduced in the special language DOLOG AKF in input modes IL, LD and FBD.

During the editing of the user program (after ”Start Entry”), you can modify the display format at will (”toggle”), under ”Presetting”. Networks are presented on screen in the display format selected, regardless of which display format they were programmed in.

There are exceptions. For example, if a network is programmed in IL and is then to be displayed in FBD, this cannot be done since it contains commands (LD,T, jumps) which cannot be shown in FBD. Another is when the width of the screen is insufficient to display all the elements (an ”AND” with more than 7 inputs, an ”OR” with more than 16 inputs, or several FBD elements cannot be displayed in LD).

Exceptional cases such as these are always presented in IL, regardless of the display format selected.

Whilst a network is being edited and reproduced, the addressing (ABS/SYM) can be changed at will, under ”Presetting”.

After input, the program is automatically examined for syntactical accuracy.

The next section is a description of the block editor for OBs, PBs, FBs, AEs and AZs (sequence control elements).

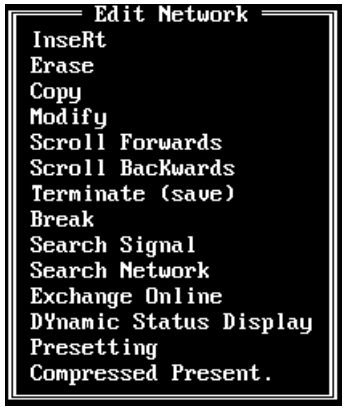
Edit block

- ”Edit”, ”Blocks”, ”Start Entry”
- ”Edit”, ”Overview”, select block, <Ctrl>+<Return>, ”Edit Block”

After defining the other settings in the pulldown menus, editing can start.

If the block you have set up is not present, it will be created and displayed in a block network .

To call the editor pulldown menu, press <Ctrl>+<Return>.



You cam also call an appropriate help text at each step with <F10>.

You can terminate the editing procedure at any time with the <Esc> key. However, you then lose any data you have entered up to that point.

Editing in various display formats

In the case of OBs, PBs, FBs, (TBs,) AEs and AZs you can program in one of three possible display modes (IL, LD, FBD) . It is possible to switch between each of the special languages at any time. There are, however, some exceptions to be aware of.

The various editors are explained on the following pages.

Certain menu lines or functions are present at several locations in the editor, or are the same in several editors, for example the ”Edit Network” menu. These positions are only described the first time they occur. The keyword register provides further guidance.

Instruction list IL

IL Editor

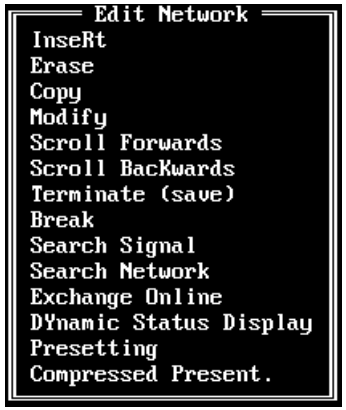
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Note If you select an existing block in the required display format, you can then edit it. Press <Ctrl>+<Return> to access the ”Edit Network” menu, which has the same structure for all display formats. The common functions are described here in IL.

You are now in the IL editor, which is described below.

You can edit in IL
a) in the pulldown menu ”Edit Network”, by using the reference characters, by selecting menu lines with the arrow keys and calling with <Return>, or by using the mouse
b) with <Ctrl>+<reference character> from a higher editor level
c) by using the special keys

a) In the pulldown menu ”Edit Network” the following functions are provided



b) If you have not selected the ”Edit Network” menu, you can carry out functions with <Ctrl>+<reference character>, e.g. <Ctrl>+<**M**> for ”**M**odify”.

c) The following special keys are valid:

<PgDn>	Scroll to next network
<PgUp>	Scroll to previous network
<Esc>	Terminate input without saving
<Home>	Modify specified network
<Ins>	Insert new network (before current one)
<Del>	Delete current network
< ↓ >	Move screen up
<↑>	Move screen down
<End>	Terminate input, translate and save

During network creation in IL, editing is carried out line by line in tabular form, i.e. certain Tab positions are reserved for the IL commands (operations), for the operands and for the 31–character IL line comments (for FBs). These can be accessed by pressing the <tab> key or the arrow keys.

In the case of OBs, PBs, FBs, (TBs), AEs and AZs, a comment of up to 31 characters can be attached to each IL line. Within comment lines, the line editor / full typewriter keyboard (including e.g. <backspace>) can be used.

Edit FBD, IL, LD / Insert network

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

You can use this ”Insert” function from the ”Edit Network” pulldown menu to insert a new network. Note that there are differences between the way networks are inserted in OBs, PBs and FBs, and the way they are inserted in AEs and AZs.

OB, PB, FB
An empty network is inserted in front of the displayed network .

There is an exception: Insertion in front of the first network of an FB is not permitted.

Insert is equivalent to the <Ins> key.

In a new network you can call further editing menus with <Ctrl>+<Return> (see ”Modify Network” in the individual input modes).
In addition to the functions described there, the following function is available at this point.

Edit IL, LD, FBD / Delete network

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Delete”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

With the ”Delete” function from the ”Edit network” pulldown menu, you can delete the indicated network.

The last remaining network in a block (block end) cannot be deleted with this function. To delete the entire block use ”Special”, ”Delete”.

Deleting is equivalent to the <Del> key.

Edit IL, LD, FBD / Copy network

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Copy”

Use this function to copy a particular network into the current block in front of the indicated network.

Exception: nothing may be copied in front of the first network of a FB.

After copying, you are in the correction mode of the editor.
When you select this function, the following pulldown menu is displayed:

- The ”source block” is the block from which the new network is to be copied. Enter <Return> space and <Return> to display a selection window.

- The ”source network” is the network to be copied. When you press the <Return> key, you can use the line editor to enter or change the number of the source network (1...999).

- The selected network is copied with ”Start Copying”. The new network receives the number of the network in which the copying is performed (”current network”).

The previous network with this number is pushed back one position.

Edit IL, LD, FBD / Modify network

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Use this function from the ”Edit network” pulldown menu to get to the correction mode of the block editor.
You may then modify the network displayed on the screen.

Modifying is equivalent to the <Home> key.

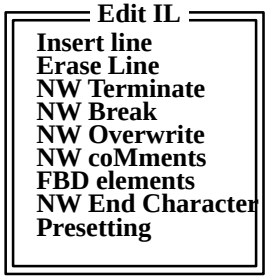
Note It is **not** possible to modify the ALSOB1. The block end network cannot be modified.

IL-Editor / Correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

From the ”Edit network”, ”Modify” menu (current block), or the ”Edit IL” menu (new block), or from the ”Action editor”, ”Modify” or the ”Action editor”, ”Modify” you move to the correction mode of the IL editor.

a) The Pulldown menu ”Edit IL” consists of the following functions:



b) If you are bypassing the ”Edit network” menu, you can carry out the functions with <Ctrl>+<reference character>, e.g. <Ctrl>+<**G**> for ”Presettin**G**”. The following keys also have an effect:

< ←>, < →>, <↑>, <↓ >	Move cursor
<Ins>	Insert a line in front of the current line (cursor on colon of current line)
<Del>	Erase line (cursor on colon of the line)
<tab>	Next input field (Tab position)
<backtab>	previous input field (Tab position)
<Ctrl>+<PgUp>	Shift image upward line by line
<Ctrl>+<PgDn>	Shift image downward line by line
<PgUp>	Shift image upward page by page
<PgDn>	Shift image downward page by page
<Esc>	Terminate input without saving

If you press the <Return> key, you can also carry out the indicated functions by means of the menu lines in the ”Edit IL” pulldown menu.

Edit IL / Insert line in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Insert line”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, present block, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Insert line”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

With this function you can insert instruction lines at any position in IL networks. You can do this by selecting the appropriate menu line and pressing <Return>, or with the reference character, or with the mouse.

Insertions are always placed in the line before the cursor.

Example:

: A M 1.1	: A M 1.1	: A M 1.1
: O E 2.12	:	: A M 5.1
: = A 3.2	: O E 2.12	: O E 2.12
: ***	: = A 3.2	: = A 3.2
	: ***	: ***
Insert before 2nd line	Line inserted	Enter instruction

Edit IL / Erase Line in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Erase Line”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, present block, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Erase Line”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Use this function to erase the current instruction line in IL networks.
The line on which the cursor is situated is erased.
When you use the special key, the cursor must be on the colon of the line to erase the line (otherwise only letters are erased).

Edit IL, LD, FBD / Terminate in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Terminate”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Terminate”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

This function terminates the correction mode of the editor, bringing correction of the network to an end, and the inserted or modified network is accepted.

Note If ”Diagnostics parameterization: ON” was chosen, you will now be taken directly to the diagnostic editor. If you do not intend to carry out diagnostics at this point, you can exit the editor with <Esc>. The diagnostic editor is described in Chapter 3.3.4.

Caution If the ”Edit network” menu is terminated with ”Break” (i.e. Cancel, Abort) or <Esc>, the amendment is ignored. ”Terminate” must also be selected for compiling and saving to the hard disk.

Edit IL, LD, FBD /Break in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Break”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Break”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

This function breaks (i.e. aborts) the network’s correction mode. Changes are ignored and the original status remains unchanged.

Break is equivalent to using the <Esc> key outside the pulldown menu.

Edit IL, LD, FBD / Network header (NW Overwrite) in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”NW Overwrite”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”NW Overwrite”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

A network header of 32 characters can be entered in the status line.

When you select this function, you are taken to the header. You can now use the line editor to enter or edit a header for the current network.

Edit IL, LD, FBD / Network comments in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”NW comments”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”NW comments”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

A comment of 15 x 60 characters maximum can be entered in a window in a network. The lines can be laid out as required. End each line with <Return>.

Press <Ctrl>+<Return> to call the following window:

Special keys provide the following functions:

< →>, < ←>, < ↑>, <↓>	Move the cursor
<Ins>	Insert mode on/off (insert character)
<Del>	Delete character under cursor
<backspace>	At start of line: transfer as many characters as possible into the previous line, otherwise: delete character to left of cursor
<Home>	Move cursor to start of line
<End>	Move cursor to line end
<Esc>	Terminate input without saving
<Return>	Within the line: new line at end of line: cursor to start of next line

Edit IL, LD, FBD / Copy comments

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”NW comments”, ”Copy Comments”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”NW comments”, ”Copy Comments”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

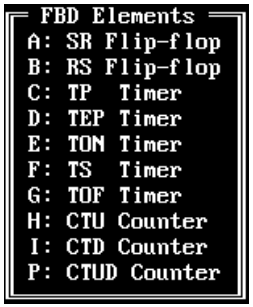
- The source block is the block from which copies are to be made. Enter <Return> space and <Return> to display a selection window.
- The source network is the network from which a comment is to be copied. When you press the <Return> key, you can use the line editor to enter or change the number of the source network (1...999).
- Use ”Start Copying” and the comment is copied from the selected network.

Edit IL / FBD Elements in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, create element, <Ctrl>+<Return>, ”FBD Elements”
- ”Edit”, ”Blocks”, ”Start Entry”, present block, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”FBD Elements”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

The elements can be called from the pulldown menu by using the reference character or selecting the menu line and pressing <Return>.
When you have selected an FBD element, the editor gives the appropriate sequence of instructions without operands (function block diagram, see page 168).

The following FBD elements can be inserted in IL:



Edit IL / NW end character in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”NW End Character”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”NW End Character
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

This function enters a network end character in the current network at the cursor position.

Caution Following lines in the network are erased.

Edit IL, LD, FBD / Presetting in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Presetting”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Presetting”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

With this function you can change the basic settings of the editor.

When the editor is in correction mode, you are provided with the following functions in the Presetting menu:

For IL:

For LD/FBD: The same points but inclusive Cursorpositioning

Edit LD, FBD / Presetting / Cursor Positioning

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Presetting”, ”Cursor Positioning”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Presetting”, ”Cursor Positioning”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Presetting”, ”Cursor Positioning”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

In automatic cursor positioning, you can toggle to set the direction in which the cursor moves between question mark fields during entry in LD/FBD.

Horizontal position:
When entering ladder address input, the cursor moves horizontally from one element to the next in a network line.

Vertical position:
When entering ladder address input, the cursor moves vertically from one element to the next in a network column.

Off position:
The cursor moves from one character position to the next on the screen.

Output monitoring

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Presetting”, ”Output monitoring”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Presetting”, ”Output monitoring”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Presetting”, ”Output monitoring”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

You can toggle between ”on” and ”off”.
When output monitoring is ”on” the address allocation is monitored during editing. If an output address already in use is entered again, a warning message is given.
This warns the programmer of multiple allocations of output assignments. The message must be acknowledged.

Edit IL, LD, FBD / Enter Symbol in SYM/COM block

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Presetting”, ”Enter Symbol in SYM/COM block”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Presetting”, ” Enter Symbol in SYM/COM block”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Presetting”, ” Enter Symbol in SYM/COM block”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

You can toggle between ”on” and ”off”.
With ”on”, you can enter a symbol and comments for an absolute address in the SYM/COM block, using either the IL, LD, or FBD editor.

An entry can be made (an input window appears):
provided the input is closed off in the editor with the <Return> key,
the input is error–free and no symbol yet exists for the hardware address.

Your input signal appears in the first line. Under it you can enter a symbol with a maximum length of Ã24 characters.The comment can be up to 40 characters long.
All input must be closed off with the <Return> key.

If a symbol for the signal has already been entered in the SYM/COM block, the input window does not appear on the screen.

When the entry has been closed off, the symbol or comment can only be changed in the ”Symbols and comments” menu.

Edit IL, LD, FBD / Scroll network forward

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>,
  ”Scroll Forward”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Current display” or ”
  Single–Scan Processing” or ”Triggered recording”, <Ctrl>+<Return>,
  ”Start Display”, <Ctrl>+<Return>, ”Scroll Forward”

After selecting this function the next network is called. If the network cannot be displayed in the current display mode (FBD/LD), there is an automatic switch to IL.

Scroll Forward is equivalent to using the <PgDn> key outside the pulldown menu.

Edit IL, LD, FBD / Scroll network backward

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Scroll Backward”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Current display” or ” Single–Scan Processing” or ”Triggered recording”, <Ctrl>+<Return>, ”Start Display”, <Ctrl>+<Return>, ”Scroll Backward”

On selecting this function, the previous network is called. If the network cannot be displayed in the current display mode (FBD/LD), there is an automatic switch to IL.

Scroll Backward is equivalent to using the <PgUp> key outside the pulldown menu.

Edit IL, LD, FBD / Terminate (save)

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Terminate (save)”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Editing is terminated when you select this function. The block which is entered is checked for syntactical correctness, compiled and stored on the hard disk.

Terminate is equivalent to the <Esc> key.

Edit IL, LD, FBD / Break

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Break”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

If this function is selected and confirmed, editing is terminated. All changes are ignored when you have acknowledged.

Break is equivalent to using the <Esc> key outside the pulldown menu.

Edit IL, LD, FBD / Network search for signal

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Signal Search”
- ”Edit”, ”Symbols and Comments”, Block, <Ctrl>+<Return>, ”Search Function”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Current display” or ” Single–Scan Processing” or ”Triggered recording”, <Ctrl>+<Return>, ”Start Display”, <Ctrl>+<Return>, ”Signal Search”

You can search for symbols and comments within a block with the search function. (The current network is not searched.)
Signal names can be entered absolutely or symbolically. When the software has found the signal it will be shown at the bottom of the screen.

You have the option to prefix with ”+” or ”-” to stipulate the search direction (forward, backward). No prefix causes forward searching.

Press the <Return> key to enable this function in the window. Use the <Esc> key to close the window again.

You can search for operands and data structures listed in the ”Pocket Guide” user instructions.

Edit IL, LD, FBD / Network Search

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Network Search”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ” Compressed Present.”, <Ctrl>+<Return>, ”Network search”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Current display” or ” Single–Scan Processing” or ”Triggered recording”, <Ctrl>+<Return>, ”Start Display”, <Ctrl>+<Return>, ”Network Search”

You can search for networks within a block with this search function.
Enter the network number and confirm with <Return>, or enter ”0” and confirm with <Return>, to display a list of all the networks in the block (showing network number and header).

Edit IL, LD, FBD / Exchange Online

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Exchange Online”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Note This function is described under ”Online”. Therefore it will not be described further at this point. It can, however, be executed here when individual blocks need to be edited. Please refer to the Chapter ”Load”, ”Exchange Online”.

Edit IL, LD, FBD / Network dynamic status display

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ” Dyn. Status Display”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

☞ **Note** The functions are described under ”Online”. Therefore they will not be described further at this point. The status display can also be executed here when individual blocks need to be edited.

The following functions can be used in dynamic status display.

The <Esc> key takes you back to editing mode.
A description can be found in the Chapter headed ”Online”, ”Dynamic Status Display”.

Edit IL, LD, FBD / Network presetting

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Presetting”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

With this function you can change the basic settings of the editor.

The following functions are available:

Select Editor Setting

Input mode	**IL**
Addressing	**ABS**
SYM Start char.	**1**
Cursor Positioning	**horizontal**
Output Monitoring	**off**
Enter Symbol in SYM/COM block	**off**

Input mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Presetting”, ”Input mode”
- ”SeTup”, ”PLC station”, ”Input mode”

This is where you set the display format in the editor. You can toggle between IL, FBD and LD.

☞ **Note** Networks which cannot be displayed in LD or FBD are automatically displayed in IL.

Addressing

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Presetting”, ”Addressing”
- ”SeTup”, ”PLC station”, ”Addressing”

- All inputs, outputs, markers, marker words, etc. can be absolutely addressed (e.g. I 2.15, M 310.17) or symbolically addressed.
- The FBD symbol of the SFB shows the SFB number in the case of absolute addressing and the SFB name in the case of symbolic addressing.

☞ **Note** If the ALD25 system is also to be visualized (VS210), all the signals that are to be visualized must be assigned a symbol.

You can toggle between the two forms of addressing.

SYM Start char.

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Presetting”, ”SYM Start char.”
- ”SeTup”, ”PLC station”, SYM Start char.”

You can select a symbol index in the range 1 ... 17.‹

This presetting is only relevant for overview, for mode LD or FBD in the editor, and for SYM addressing with symbol names longer than 8 characters..

Enter the selection accompanied by the part of the symbol name that is to be displayed in LD or FBD.

Display: 8 characters with greater–than or less–than signs

Example: symbol name = SYMBOL_GREATER_THAN_EIGHT
SYM Index 1: SYMBOL_G>>
SYM Index 9: REATER_T>>
SYM Index 17: >>HAN_EIGH

In correction mode, the displayed short form symbol can be selected with the cursor keys and then displayed in full in a window by pressing <Return>.

Ladder diagram LD

LD Editor

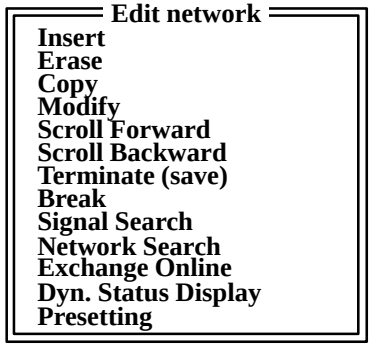
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Note You can select an existing block in the desired display format and then edit it further. Press <Ctrl>+<Return> to access the ”Edit network” menu, which has the same structure for all display format. The common functions are described under IL.

You are now in the LD editor, which is described below.

In LD you can edit
a) in the ”Edit network” pulldown menu by using the reference characters, or by selecting the menu line with the arrow keys and calling with <Return>, or by using the mouse,
b) or with <Ctrl>+<reference character> from the higher editor level.

a) The following functions are available in the ”Edit network” pulldown menu:



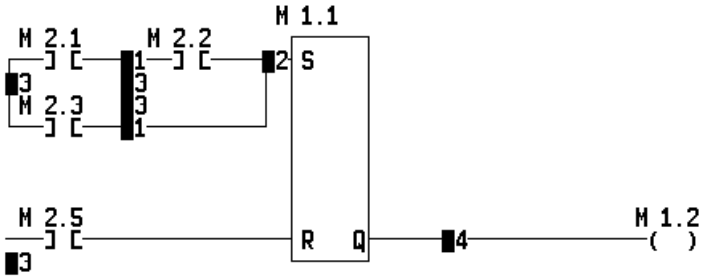
b) If you have not selected the ”Edit network” menu, you can carry out the functions by pressing <Ctrl>+<reference character>, e.g. <Ctrl>+<**M**> for ”**M**odify”. The following keys also have an effect:

<PgDn>	Scroll to the next network
<PgUp>	Scroll to the previous network
<Esc>	Break off input without saving
<Ins>	Insert new network
<Del>	Erase network
<Home>	Modify set network
< ↑>	Move screen up
<↓ >	Move screen down
<End>	Terminate input and save entry

You can also press <Ctrl>+<Return> and select the functions listed above from the menu lines of the ”Edit network” pulldown menu that is displayed at bottom right of the screen.

When programming the network in LD, it must be noted that elements can only be entered at specific cursor positions.

Positions marked with boxes and numbers must be accessed with the arrow keys. The significance of the numbers is explained after the figure.



Actions in input mode:

1: Input S,O,C: contact inserted in accordance with input
Input FBD elements: no action, invalid input position

2: Input S,O,C: contact inserted in accordance with input before path cross–connection
Input FBD elements: no action, invalid input position

3: Input S,O: new OR path created, contact inserted
Input FBD elements: no action, invalid input position

4: Input S,O,C: contact inserted in accordance with input
Input FBD elements: block inserted in accordance with input

Actions in overwrite mode:

1: Input S,O,C: contact overwritten in accordance with input
Input FBD elements: no action, invalid input position

2: Input S,O,C: no action
Eingabe FUP-Elemente: block overwritten in accordance with input (when the type is the same)

3: Input S,O,C: no action
Input FBD elements: no action

4: Input S,O,C: contact inserted in accordance with input
Input FBD elements: block inserted in accordance with input

The cursor must be on the circuit diagram to edit.

Insert:	To the right of the cursor
Insert FBD element:	Cursor must be on first character of contact (spread or activate insert mode first)
Erase:	To the right of the cursor
Erase FBD element:	Elements of the same type can be exchanged if the cursor is located on the first input (path)

For address input, it is possible to use automatic cursor positioning in the ”Pre-setting” menu.

Address input:

New input:
If the cursor is on an address field and a letter or number is entered, then an entire address identifier is deleted and a new entry requested.

Correction:
If the cursor is on an address field and the <Return> key is pressed, the entire address identifier is highlighted and you can then use the cursor to select the place to be corrected.

Insert LD element:	To the right of the cursor
Insert FBD element:	Cursor must be on first character of contact (spread first if necessary)
Erase LD element:	To the right of the cursor
Erase FBD element:	Elements of the same type can be overwritten when the cursor is on the first input (path).

For address input, it is possible to use automatic cursor positioning in the ”Presetting” menu.

Address input begins at the first question mark.

LD Editor / Correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

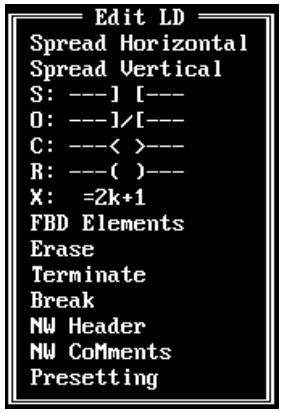
From the ”Edit network” menu (present block) or the ”Edit LD” menu (new block after inputting the first element), you can access the correction mode of the LD editor.

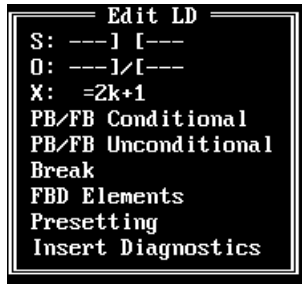
You can then modify the network displayed on the screen.

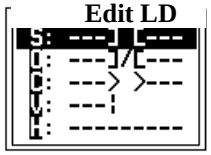
There is an exception: The block end network cannot be modified.

Modify is equivalent to the <Home> key.

a) There are 3 ”Edit LD” pulldown menus that appear during editing:







b) You can carry out the functions by pressing <Ctrl>+<reference character>, e.g. <Ctrl>+<**F**> for "**F**BD–Elements". The following keys also have an effect:

< ←>, <→>, <↑>, < ↓>	Move the cursor
<Del>	Delete element
<PgUp>	Move image up line by line
<PgDn>	Move image down line by line
<Ctrl>+<PgDn>	Move image up page by page (scroll forward)
<Ctrl>+<PgUp>	Move image down page by page (scroll backward)
<Character>	Change contact address
<Esc>	Break off without saving
<Ins>	Spread vertically
<Ctrl>+<→>	Spread horizontally
<Ctrl>+<I>, <F9>	Toggle insert mode / overwrite mode
<F1>	–] [– + –( )– (only allowable as first input in network)
<F2>	–] [– (normally open contact)
<F3>	–]/[– (normally closed contact)
<F4>	–| (Zurückführung im Parallelpfad)
<F5>	2k+1 (exclusive OR)
<F6>	–> >– (connector)
<F7>	——— (continuation in the parallel path)

By pressing<Ctrl>+<Return> or <Return>, you can also carry out the functions listed above using the menu lines in the various "Edit LD" pulldown menus.

Edit LD / Horizontal spread in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Spread Horizontal”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Spread Horizontal”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

The network to the right of the cursor is spread horizontally. It is not possible to spread before the first input (left edge of the screen).

The network is optimized by inserting FBD elements.

In the insert mode, it is not necessary to spread.

Spread Horizontal is equivalent to <Ctrl>+<→>.

Edit LD / Vertical spread in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Spread Vertical”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Spread Vertical”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

It is necessary to spread vertically before inserting a new parallel path. The cursor must be situated below the horizontal series path. Spreading takes place above the cursor position.

In the insert mode, it is not necessary to spread.

Spread Vertical is equivalent to using <Ins> outside the pulldown menu.

Edit LD / Normally open contact in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Normally open contact”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return> ”Modify”, <Ctrl>+<Return>, ”Normally open contact”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol ⊣⊢
Reference character <S>

This function serves to edit a normally open contact to the right of the cursor position:

a) in a serial path:
In a new network the output is inserted at the same time as the function is selected.

b) in a parallel path:
With this function, a normally open contact, as the first element of a parallel path, can be edited. For this purpose, the cursor must be at least 3 lines below an element, or insert mode must be active.

Normally open contact is equivalent to the <F2> key.

Edit LD / Normally closed contact in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Normally closed contact”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Normally closed contact”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol ⊣/⊢
Reference character <O>

This function serves to edit a normally closed contact to the right of the cursor:

a) in a serial path:
   In a new network, selecting this function is accompanied by insertion.

b) in a parallel path:
   A normally closed contact can be edited as the first element of a parallel path using this function. For this purpose the cursor must be at least 3 lines below an element, or insert mode must be active.

Normally closed contact is equivalent to the <F3> key.

Edit LD / Connector in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Connector”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Connector”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol –> >–
Reference character <C>

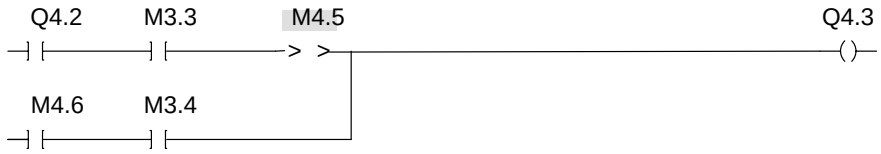
With the connector, a signal is laid on an intermediate marker or output without register initialization.

Inserting optimizes the network:

Connector is equivalent to the <F6> key.

Example

NW1



In NW1 a connector has marker M4.5 assigned.

Edit LD / Output in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Output”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Output”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol
Reference character: <R>

This function is used for inserting an output:

a) In a new network, before the <Return> key is pressed (before the pulldown menu) an output can be edited with <Ctrl>+<R> or <F1>. Simultaneously, the first normally open contact is inserted.

b) in an existing network, an output signal is multiplied by using this function. The cursor must be at least 3 lines below the previous output, or insert mode must be active.

Output is equivalent to the <F1> key.

Edit LD / Exclusive OR block in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Exclusive OR”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Exclusive OR”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol =2k+1
Reference character <X>

This function is used to edit an exclusive OR block (max. 46 inputs).

Two inputs:	if the block has two inputs, the output is activated if a ”1” signal or ”0” signal is applied to both of the inputs (exclusive OR).
Several inputs:	if the block has more than two inputs, the input is activated when an odd number of inputs carry a ”1” signal or ”0” signal (”odd element”).

Insertion optimizes the network structure.

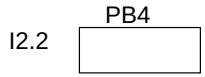
Exclusive OR is equivalent to the <F5> key.

Edit LD, FBD / Conditional PB/FB

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”PB/FB Cond’al”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”PB/FB Cond’al”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

With this function you call a block conditionally. By this method, whether a block is called depends on the result of a signal address.

e.g.



PB 4 is only called when I 2.2 = 1

You must personally enter the name of the block to be called. Enter the condition address to the left of the block.

The call condition can also be negated.

FB: The user FB must be declared before calling. (see function block)
SFB: Standard FBs can be called directly, either by SFB number (e.g. ”SFB101”) or by name.
PB: These can be called (even in advance of programming)
AZ: These can be called (even in advance of programming)
OB: These cannot be called

Edit LD, FBD / Unconditional PB/FB

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”PB/FB/SFB Uncond’al”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”PB/FB/SFB Uncond’al”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

With this function you call a block unconditionally, i.e. it is run through on every scan.
You must personally enter the name of the block to be called. Blocks are unconditiontally entered in accordance with their function, PB/FB/SFB/AZ (see above), which means that they will always be run through.

Edit LD, FBD / FBD elements in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”FBD Elements”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”FBD Elements”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

The elements can be called from the pulldown menu with the reference characters or by selecting the menu lines and pressing <Return>.

The entries under LD editor and ”Edit”, ”Blocks” are valid for the input of FBD elements.

In addition:
In LD you can insert further FBD elements only to the right of existing elements. The network is optimized when the element is inserted.

Exchanging FBD elements of the same type (at the same place) is permissible. The cursor must be on the path of the first entry for this.

A list of the permissible FBD elements can be found in the section headed Display Format (function block diagram, page 168).

Edit LD / Erase in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Erase”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Erase”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

This function is used for erasing elements from a network. All LD elements to the right of the cursor are erased in the LD editor. When dealing with FBD elements, the cursor must be situated on the first input path in order to erase.

Erase is equivalent to using <Del> outside the pulldown menu.

Edit LD / Parallel path

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, parallel path
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, parallel path
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

This and the following functions take effect when you want to edit a second path parallel to the first path in a network.
The pulldown menu can be called with <Ctrl>+<Return>.

The menu remains available until the termination of the parallel path. You can select the following functions:

Edit LD / Normally open contact in parallel path

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, parallel path, normally open contact
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, parallel path, normally open contact
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol ⊣⊢
Reference character: <S>

This function serves to insert a serial normally open contact in a parallel path.

Normally open contact in parallel path is equivalent to the <F2> key.

Edit LD / Normally closed contact in parallel path

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, parallel path, normally closed contact
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, parallel path, normally closed contact
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol -|/|-
Reference character: <O>

This function serves to insert a serial normally closed contact in a parallel path.

Normally closed contact in parallel path is equivalent to the <F3> key.

Edit LD / connector in parallel path

- ”Edit”, ”Blocks”, ””””Start Entry””, create element, <Return>, parallel path, connector
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol -> >- Reference character: <C>

This function serves to insert a serial connector in a parallel path.

The network is optimized when the element is inserted.

Connector in parallel path is equivalent to the <F6> key.

Edit LD / Cross–connection of the parallel path

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, parallel path, cross–connection
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, parallel path, cross–connection
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol —┤
Reference character: <V>

With this the open parallel path is closed at the cursor position.

Caution It is only permissible to insert the symbol if the parallel path has been created correctly, otherwise the wrong path is deleted.

Cross–connection of the parallel path is equivalent to the <F4> key.

Edit LD / Continuation in parallel path

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, parallel path, continuation
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, parallel path, continuation
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol ——
Reference character: <H>

The path is continued without contacts.

Continuation of the parallel path is equivalent to the <F7> key.

FBD function block diagram

FBD Editor

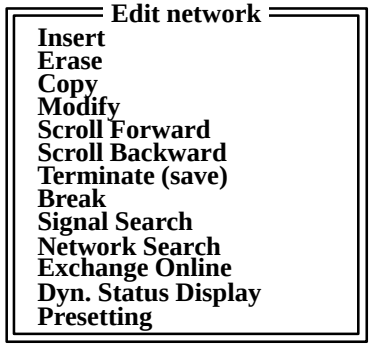
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Note You can select an existing block in the required display format and then edit it. Press <Ctrl>+<Return> to access the ”Edit network” menu, which has an identical structure for all display formats. The common functions are described under IL.

You are now in the FBD editor, which is described below.

You can edit in FBD
a) in the ”Edit network” pulldown menu by using the reference characters, or by selecting menu lines with the Arrow keys and calling with <Return>, or by using the mouse,
b) or with <Ctrl>+<reference character> from the higher editor level.

a) The following functions are available in the ”Edit network” pulldown menu:



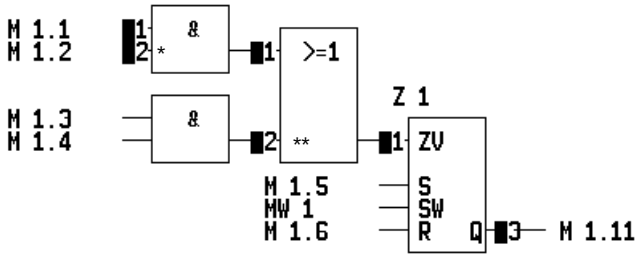
b) If you have not selected the ”Edit network” menu, you can carry out the functions by pressing <Ctrl>+<reference character>, e.g. <Ctrl>+<**M**> for ”**M**odify”. The following keys also have an effect:

<PgDn>	Scroll to next network
<PgUp>	Scroll to previous network
<Esc>	Break off input without saving
< ↑>	Move screen up
< ↓>	Move screen down
<Home>	Edit set network
<Ins>	Insert new network
<Del>	Delete network
<End>	Close off input and save.

You can also press <Ctrl>+<Return> and select the functions listed above from the menu lines of the ”Edit network” pulldown menu that is displayed at bottom right of the screen.

When programming a network in FBD it should be noted that elements can only be input at specific cursor positions.
These positions must be accessed with the arrow keys:

The positions marked with boxes and numbers must be accessed with the arrow keys. The significance of the numbers is explained after the figure.



* no insertion of input here
** no overwriting of input here

Actions in input mode:

1: Input A,O,X: Block inserted in accordance with input
Input N,I: no action
Input FBD elements: block inserted in accordance with input

2: Input A,O,X: Block inserted in accordance with input
Input N,I: Block input inserted in accordance with input
Input FBD elements: block inserted in accordance with input

3: see 1

Actions in overwrite mode:

1: Input A,O,X: Block overwritten in accordance with input
Input N,I: Block input overwritten in accordance with input
Input FBD elements: block overwritten in accordance with input,
(for the same type)

2: Input A,O,X: no action
Input N,I: Block input overwritten in accordance with input
Input FBD elements: no action

3: Input A,O,X: Block inserted in accordance with input
Input N,I: no action
Input FBD elements: block inserted in accordance with input

Insert:	To the right of the cursor; the cursor must be on the path of the input.
Insert FBD element:	To the left of the cursor; the cursor must be on the output of the path
Delete:	To the right of the cursor; the cursor must be on the path of the input.
Deleting an FBD element:	Elements of the same type can be exchanged when the cursor is on the first input (of the path) and overwrite mode is active.

For address input, it is possible to use automatic cursor positioning from the ”Presetting” menu.

Address input:

New input:
If the cursor is on an address field and a letter or number is entered, then an entire address identifier is deleted and a new entry requested.

Correction:
If the cursor is on an address field and the <Return> key is pressed, the entire address identifier is highlighted and you can then use the cursor to select the place to be corrected.

FBD Editor / Correction mode

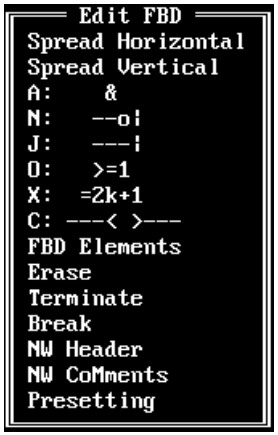
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

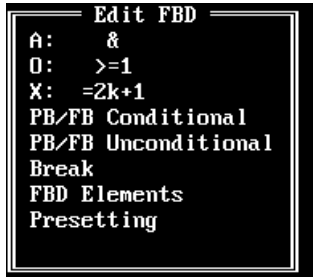
From the ”Edit network” menu (present block) or the ”Edit FBD” menu (new block after inputting the first element), you can access the correction mode of the FBD editor.You can then modify the network displayed on the screen.

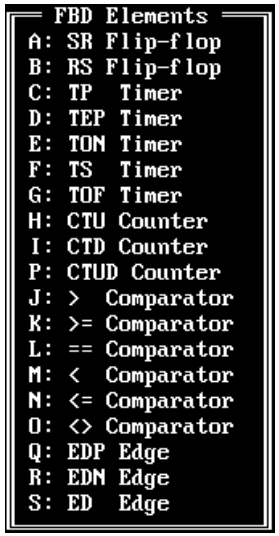
There is an exception: The block end network cannot be modified.

Modify is equivalent to the <Home> key.

a) There are 3 ”Edit FBD” pulldown menus and an ”FBD Elements” menu that appear during editing:







b) You can carry out the functions by pressing <Ctrl>+<reference character>, e.g. <Ctrl>+<**F**> for "**F**BD Elements". The following keys also have an effect:

<←>, <→>, < ↑>, <↓>	Move the cursor
<Ctrl>+< → >	Spread Horizontal
<Del>	Delete element
<PgUp>	Move image up line by line
<PgDn>	Move image down line by line
<Ctrl>+<PgDn>	Move image up page by page (Scroll Forward)
<Ctrl>+<PgUp>	Move image page by page Scroll Backward)
<Character>	Change contact address
<Esc>	Break off without saving
<Ins>	Spread Vertical
<Ctrl>+<→>	Spread Horizontal
<End>	Terminate correction
<Ctrl>+<I>, <F9>	Toggle insert mode / overwrite mode
<F1>	& (AND)
<F2>	⊣/⊢ (negated input)
<F3>	⊣⊢ (input)
<F4>	>=1 (OR)
<F5>	=2k+1 (exclusive OR)
<F6>	–> >– (connector)

If you press <Return>, you can also carry out the functions listed above with the aid of the menu lines in the various "Edit FBD" pulldown menus.

Edit FBD / Spread Horizontal in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Spread Horizontal”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Spread Horizontal”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

The network to the right of the cursor is spread horizontally.The network is optimized by inserting FBD elements.

In insert mode, spreading is not necessary.

Spread Horizontal is equivalent to using <Ctrl>+<→> outside the pulldown menu.

Edit FBD / Spread Vertical in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Spread Vertical”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Spread Vertical”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Vertical spreading is necessary in order to insert further inputs into an FBD symbol.

The cursor must be below the first input path of the FBD symbol.

The network is optimized when the element is inserted.

In insert mode, spreading is not necessary.

Spread Vertical is equivalent to using <Ins> outside the pulldown menu.

Edit FBD / AND block in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”AND”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”AND”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol &
Reference character <A>

This function is used to edit an AND block (max. 46 inputs).

The network is optimized when this element is entered.

AND is equivalent to the <F1> key.

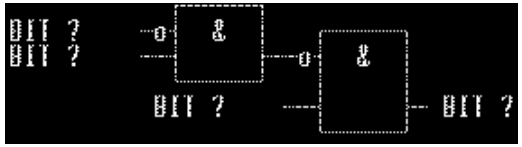
Edit FBD / Negated input in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Negated input”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Negated input”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol

Reference character <N>

This function serves to negate an input. Outputs cannot be negated. It is possible to negate inputs that are also outputs. It is possible to insert an additional negated input or to overwrite inputs.



The network is optimized when this element is entered.

Negated input is equivalent to the <F2> key.

Edit FBD / Input in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Input”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Input”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol
Reference character <I>

This function serves to edit an input. An additional input can be inserted or negated inputs can be overwritten.

The network is optimized by insertion or overwriting of this element.

Input is equivalent to the <F3> key.

Edit FBD / OR block in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”OR”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”OR”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol >=1
Reference character <O>

This function is used to edit an OR block (max. 46 inputs).

The network is optimized when this element is entered.

OR is equivalent to the <F4> key.

Edit FBD / Exclusive OR block in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Exclusive OR”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Exclusive OR”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol =2k+1
Reference character <X>

This function is used to edit an exclusive OR block (max. 46 inputs).

Two inputs: if the block has two inputs, the output is activated if a ”1” signal or ”0” signal is applied to both of the inputs (exclusive OR).

Several inputs: if the block has more than two inputs, the output is activated, if a ”1” signal or ”0” signal is applied to an odd number of inputs (”odd element”).

Insertion optimizes the network structure.

Exclusive OR is equivalent to the <F5> key.

Edit FBD / Connector in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Connector”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Konnektor”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

Symbol –> >–
Reference character <C>

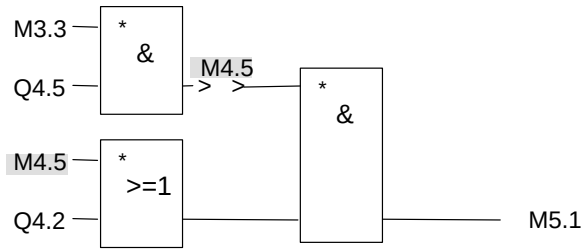
With the connector, a signal is laid on an intermediate marker or output without register initialization.

The network is optimized when the element is inserted.

Connector is equivalent to the <F6> key.

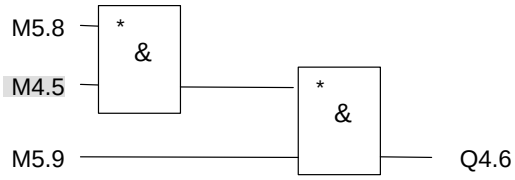
Example

NW1



In NW1 a connector has marker M4.5 assigned. The content of the marker is used in the same scan as input of the OR block.
In addition, the marker M4.5 is used as input of the first AND block in the next network.

NW2



Edit FBD / Erase in correction mode

- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Insert”, <Ctrl>+<Return>, ”Erase”
- ”Edit”, ”Blocks”, ”Start Entry”, <Ctrl>+<Return>, ”Modify”, <Ctrl>+<Return>, ”Erase”
- ”Edit”, ”Overview”, select block, <Return>, ”Edit Block”,..

This function is used to delete an element from a network or gate inputs.

Deleting an element:
The cursor has to be in the upper left corner of the block.

Deleting several continuous elements:
The cursor must be on the output of the element sequence to be deleted.

Deleting gate inputs:
The cursor must be on an output occupied by a signal.

The network is optimized when this function is executed.

Erase is equivalent to using the <Del> key outside the pulldown menu.

Block

- ”Edit”, ”Blocks”, ”Block”
- ”Load”, ”Echange Online”, ”Block”
- ”Load”, ”Exchange Online”, ”Block”

Here you can enter the block (OB1-999, PB1-999, FB1-999, AE1-255, AZ1-999, KB1-255) that you want to process.

If you would like to see a display of the blocks present in the station, press the space bar and then the <Return> key. You can then select a block in the list, using the arrow keys and the <Return> key.

Organization block OB
Organization blocks (OBs) determine the order in which the individual blocks should be processed.

Program blocks, function blocks and sequence blocks PB, FB, SFB and KB, and the cyclical actions are called and processed by the OB in the required order.

The following particulars apply to OB editing:

- ❒ OB1 must always be edited. This determines the framework for the **complete** user program.
- ❒ OB3 handles the linking of sequence controls.
- ❒ There are 4 OBs for interrupt processing
- ❒ There are 5 OBs for automatic control engineering
- ❒ The OB is processed by cyclical scanning. Each scan starts by processing network 001 and ends when the last network present in the OB is processed.
- ❒ PB calls, FB calls, AZ calls and possibly the user program are contained in consecutively numbered networks beginning with network 001.
- ❒ Each network can contain only one PB, FB or AZ (except in IL).
- ❒ It calls, conditionally or unconditionally, PBs, FBs, SFBs and AZs. You can only enter an FB call if you have previously edited the block.

- It can also contain program parts in IL, LD or FBD.

Program block PB
In general, a program block contains parts of the user program which technically belong together, e.g. one of x different machines.

- A PB consists of a sequence of networks, consecutively numbered starting with network 001.

- It can call PBs, FBs, SFBs and AZs conditionally or unconditionally. You can only enter an FB call when you have previously edited the block.

- The same PB can be called several times.

- It contains program parts in IL, LD or FBD.

- In OB (or another PB), a block that is to be called is displayed as a square (in LD/FBD) in the network. The PB number is above the square. For conditional PB calls, the signal address of the call condition is to the left of the square. The call is also possible in FB.

- A PB that is not called from any place is never processed.

Function block FB
FBs are used to create frequently recurring program parts. They are parameterizable subroutines, i.e. an FB can be called and parameterized many times at various points.

- It can call PBs, FBs, SFBs and AZs conditionally or unconditionally. You can only enter an FB call when you have previously edited the block.

- The same FB can be called many times.

- It contains program parts in IL, LD or FBD.

FB1 ... FB999: user function blocks
In order to optimize time critical processes, you should create your own solutions whenever possible. For this purpose, you can program up to 999 function blocks.

You must be aware of the difference between a function block and a function block call. An FB contains part of a user program. An FB call ensures that, during runtime, the FB will be processed at exactly the time when the appropriate call is encountered in the user program. For the processing of the block, the formal operands of the FB are replaced with the declared actual operands. An FB that is not called from any place is also never processed.

Structure
The function block program consists of the Declaration part and the instruction part (IL, LD, FBD). The declaration part is in the first network of an FB. The instruction part follows, in subsequent networks.

Declaration part

The declaration part contains the name of the function block and a list of the formal operands, with declaration of type.

After selecting an FB in the editor, the following mask appears in network 1:

```
NAME:
     <IDENTIFIER  DESIRED TYPE  ATTRIBUTE>
     :***
```

Creating the declaration part

Step 1 Enter the name of the function block in the first line and confirm with <Return>.

Step 2 Insert a blank line with <Ctrl>+<E>.

Step 3 Insert the first formal operand in the first blank line behind the colon (max. 6 letters/numbers; the first position must always be a ketter but the other positions are freely definable).

Step 4 Call the selection window by entering a space in the ”required type” column.

Step 5 Choose the required type with the arrow keys and confirm with <Return>.

Step 6 In the column ”Attribute”, you determine whether the formal operand is to be used as an input parameter (I) or as an output parameter (O) of the block. Determine whether the formal operand should be stored in the signal memory (no entry) or not (F). (With <backspace> you can delete existing entries.)

Step 7 Carry out Step 2 bis Step 6 for all other lines.

Step 8 Save the correct entries with <Ctrl>+<B>.

An example of a declaration part follows.

```
NAME:   EX1
        <IDENTIFIER  REQUIRED TYPE   ATTRIBUTE>
 IDENT  :  OP1             BIT       I
 IDENT  :  OP2             BIT       I
 IDENT  :  OP3             BIT       I
 IDENT  :  OFF1            BIT       O
     :  ***
```

Identifier
A parameter name with max. 6 characters is to be entered here

Required type
In ”Required types”, you can select a desired setpoint type from the menu lines by pressing the space bar. The setpoint type can also be entered directly.

Meanings of the required setpoint types:

BIT	Input, output, marker
BYTE	Byte marker
WORD	Word marker
DWORD	Doubleword marker
GWORD	Floating point word marker
T	Timer (time marker)
C	Counter
B2	Bit string width max. 2 bits in a word
B8	Bit string width max. 8 bits in a byte
B16	Bit string width max. 16 bits in a word
B32	Bit string width max. 32 bits in a doubleword
ANZ	Bit string width (see also page 121)
NODE	Node number (number from equipment list) for this required type; the module type of the node must also be entered in the attribute field. If the 3rd character in the attribute field is blank (space), a selection window is displayed and you can use it to choose the module type. The module type can also be entered directly.
STRUCT	Data structure; a selection window appears; it must have been previously edited in the data structure editor.
PROC	Another block (OB or PB); a selection window appears
V	Constant
POINTER	Pointer

Attribute
If the attribute field is blank, the system makes a suggestion which can be accepted by pressing <Return>. Permissible entries are:

I	read-only identifier
O	read-write identifier
F	far addressing (no signal memory address)
	the presetting near (signal memory) is not displayed

An entry from the attribute column (I/O) has the effect of stating whether the parameter should appear in the display as input on the left edge of the block or as output on the right edge of the block. The option ”F” indicates that the parameter is not in the signal memory.

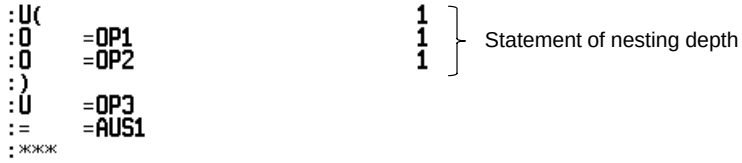
An appropriate representative of each of the two groups can be declared.

The declaration part closes with”***” (network end).
When changes are made later, it is possible to delete or insert parameters. The formal operand name itself can be changed at any time. Whenever changes are made, it is essential that all calls be revised.

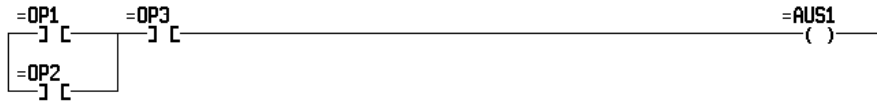
Instruction part
In the instruction part, the program is in IL, LD or FBD. This establishes algorithmic correlations between the formal operands in the declaration part. The names of the formal operands in the instruction list must always be preceded by an ”=” sign. In some cases, to the right of the instruction list is a number which specifies the nesting depth of the line in question. The instruction list always ends with ”:BE” (block end).

The figures below show sections of the instruction part of the above–mentioned function block ”EX1” in IL, LD and FBD.

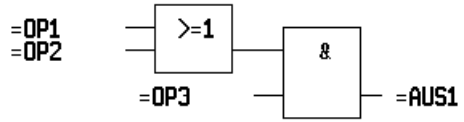
IL



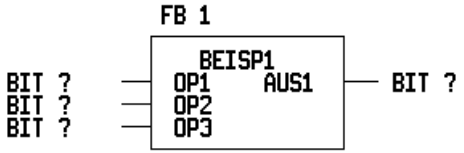
LD



FBD



Call



An FB can be called from another FB, PB or OB.
The same FB can be called many times.
When the FB is called from a PB or OB in LD/FBD, it is displayed as a square in the network. In the square, its description is once again displayed, in short form. In the square on the left are the input formal operands,and outside, on the left, the actual operands. Above is possibly another condition. In the square on the right are the output formal operands, and outside, on the right, the actual operands.

In IL, the call is carried out with operations BA or BAB.

If formal parameters are changed, deleted or added during editing of the declaration part of a function block, then all function block calls have to be reprogrammed. To find the places in the user program where the function block calls for the function block in question are located, use the global cross–reference list.

You can call a function block in the instruction part of another function block (nesting, recursion). The function block called in this way can contain the formal operands of the function block from which to call as actual operands. Thus, during user program run time, actual operands can be passed on, even as far as the deepest nesting level (10).
Program blocks tested by the user and found to be appropriate can easily be converted to function blocks FB1...FB999 (see PB).

(Standard function blocks SFBs)
SFB1 ... SFB999: Standard function blocks
are provided by Dolog AKF. The SFBs have various performance features. You can use them as often as you like, you need only parameterize them for your purposes.
Basically, there are 999 possible SFBs.

To parameterize SFBs you can use, for instance, data structures. A description of how to proceed when using the AEG predetermined data structures is presented stepwise below.

- First look at the tables in ”A250 Pocket Guide” to find out which data structures your SFB possesses.

- For each SFB call, assign symbols and, when appropriate, initial values, to all elements (e.g. APOA**1.1** to APOA**1.13**) in a copy of the data structure (e.g. APOA**1**) in ”Symbols and Comments”. If you have imported the above files, you will be offered suggestions for the first sample. You can overwrite the presettings at any time.

- For each SFB, enter a copy (e.g. APOA**1**) of the associated data structure.

Network number

- ”Edit”, ”Blocks”, ”Network number”

At this point enter the network (1-999) of the block that you wish to process. Press <Return>; you can then enter or change the number with the line editor (typewriter keyboard).

Display Formats IL, LD, FBD (Input mode)

In the display type (see also ”input mode”), you can choose between instruction list (IL), ladder diagram (LD) and functional block diagram (FBD). The individual special languages in the Dolog AKF block editor are described in more detail below:

Instruction list IL
The instruction list is a standardized alphanumeric display format that conforms to DIN 19239.

An instruction list is made up of several instruction lines. It consists of a section for jump marks (ahead of the colon), a colon, the operation and the operand

```
    :   operation      operand
```

OBs, PBs, FBs, AEs, AEs and TBs can be programmed in IL.

Rules for constructing an instruction list

network start / end

Example:

```
:   A..., O..., L...
or LA...
:   ***
```

Each network must start with the operations A, O (for logic) or L, LA (for arithmetic) and terminate with " *** " .

Switch–on marker

The switch–on marker SM39 (startup marker) acts as an initiating pulse. The valency of the switch–on marker is only =1 for the first two program cycles .

Utilization: For initializing the actual values of timers and counters in hot restart operating mode when powering–up or on program start. For details of how to program memories with preferred state after power–up, see "Memory".

marker bit

Example

```
Network 001
:   A   I2.1
:   A   I2.2
:   O   I2.3
:   =   M5.1
:   ***
Network 002
:   A   I2.4
:   A   M5.1
:   =   Q3.1
```

Markers (binary intermediate results) save the results of logical operations for use elsewhere in the program. They have to be defined before the scan. For nesting, it is advisable to work with markers or bracket operations.

Bracket operations (only in IL)

Example

```
:   A   (          Bracket operations can also be used in place of
:   A   I2.1       markers.
:   A   I2.2       After an instruction ”Open brackets”, just as at the start of
:   O   I2.3       the program, the next instruction has to begin with A or O.
:   )              The maximum nesting depth for bracket operations is
:   A   I2.4       13. The number of ”Open brackets” operations
                   must be the same as
:   =   Q3.1       the number of ”Close brackets” operations.
:   ***
```

Block end

The last network of each block contains only the ”block end”:

```
:   BE
```

Logic functions

Order of logic operations

Example

```
:   A   I2.1        When logical operations are being calculated, the rule ”AND
:   A   I2.2        before OR” is applied, as in Boolean algebra,
:                   i.e. superfluous brackets need not be entered.
:   O   I2.3        The accompanying IL obeys this rule
:   O   I2.4        (I2.1 ∧ I2.2) ∨ I2.3 ∨ I2.4 = Q3.1
:   =   Q3.1
:   ***
```

Initializing operations

Instruction with initialization

Example

```
:   A   I2.1        With operations such as S..., R... the register is reset
:   S   M5.19       (initialized), and with it the result of the logic operation.
```

The result cannot be

```
:   A   I2.2        directly used for the next operation . Consequently, the
:                   following instruction  must always begin with the operation
:   ***             A or O. This also applies to the instructions
                    A(, O(, X(, UN(, ON(, XN(. Instruction without
:   A   M5.4        initialization.
```

The result of logic operation AM5.4, AM5.5 is not initialized in the case

```
:   A   M5.5        of assignment ”=”, i.e. it is available for further assignment
:   =   M5.6        (multiple assignments are possible).
:   =   M5.7
:   =   M5.8
:   A   I2.7        Before the next ”AND” operation is executed, the register
:                   is first initialized, i.e. erased, and therefore the previous
:   ***             result of logic operation can no longer be used.
```

Connector

With the connector, =C, a signal is placed on an intermediate marker or register. The register is not initialized (reset), so networks can be saved. The marker type must be congruent with the operands of the other instructions in the command sequence.

Examples

Logic

```
:   A   M3.2
:   A   M4.5
:   =C  M3.5
:   O   M4.8
:   =   Q3.7
:   ***
```

Arithmetic

```
Network 001
:   L     V12
:   ADD   MW1
:   =C    MW30
:   SUB   MW2
:   =     MW44
:   ***

Network 002
:   L     MW30 Intermediate result from NW001.
:   ADD   MW15
:   =     MW45
:   BE
```

Save operations
The sequence of IL commands must be strictly followed. The preferred state after power–on (startup marker) SM2 or additional instructions before the ”=” assignment are optional.
In cases where no additional instructions precede the ”=” assignment, the marker M* can also be replaced with Q3.*t in the examples mentioned below.

Example: SR memory, dominant reset, preferred state after power–on 1

```
:   A   (
:   O   I2.1      With effect from the first program cycle, system marker SM39
:   O   SM39      has a ”1” signal (automatically reset later), and therefore
:   )             preferred state after power–on 1.
:   S   Q3.1
:   A   I2.2      The reset condition is processed after setting, hence
:   R   Q3.1      dominant reset.
:   =   Q3.1
:   ***
```

Example: RS memory, dominant set, preferred state after power–on 1

```
:   A   I2.2
:   R   Q3.2
:   A   (
:   O   I2.1      With effect from the first program cycle, system marker SM39
:   O   SM39      has a ”1” signal (automatically reset later), and therefore
:   )             preferred state after power–on 1.
:   S   Q3.2      The set condition is after the reset condition, hence
:   =   Q3.2      dominant set.
:   ***
```

Note Examples for dominance, preferred state after power–on 0 and preferred state after power–on 1 can be found on page 167.

Counters

Please adhere strictly to the command sequence C, S, L and R in order to avoid syntax errors. There can be additional instructions between commands.

CU, counter up 0 ... 32 767

Example:

```
:   A   I2.1        Counting pulse input (count with 0 → 1 edges on I2.1).
                    With the first edge, the counter output is set to ”1”.
:   CU  C10         C10, up counter.
:   A   I2.2        With ”1” signal on I2.2, the setpoint value is loaded
                    with V30.
:   S   C10
:   L   V30
:   A   I2.3        Reset input (actual value and counter output are set to ”0”
:   R   C10         with ”1” signal on I2.3).
:   =   Q3.1        When actual value = setpoint value: ”0” signal on Q3.1.
```

CD, counter down 0 ... 32 767

Example:

```
:   A   I2.1        Counting pulse input (count with 0 → 1 edges on I2.1).
                    With the first edge, the counter output is set to ”1”.
:   CD  C12         C12, down counter.
:   A   I2.2        With ”1” signal on I2.2, the setpoint value is loaded
:   S   Z12         with CSW12.
:   L   CSW
:   A   I2.3        Reset input (actual value and counter output are set to ”0”
:   R   C12         with ”1” signal on I2.3).
:   =   Q3.2        When actual value = setpoint value: ”0” signal on Q3.2.
```

C+, C-, counter up, down -32 768 ... +32 767
It is possible to count both up and down with this counter.
Example:

:	A	I2.1	Counting pulse input (count up with 0 → 1 edges on input I2.1).
:	C+	C12	C12, count up
:	A	I2.2	counting pulse input (count down with 0 → 1 edges on input I2.2).
:	C-	C12	C12, count down
:	A	I2.3	With ”1” signal on I2.3, the setpoint value CWS13
:	S	C12	is loaded.
:	L	ZSW	
:	A	I2.4	Reset input (actual value and counter output are set to ”0”
:	R	C12	with ”1” signal on I2.4).
:	=	Q3.2	When actual value = setpoint value: ”1” signal on Q3.2.
:	***		

Timers

In the case of IL input, the function and operating time characteristics of timers are exactly equivalent to the FBD elements of the FBD or LD editor. There are 5 timer functions available. The associated timing diagrams can be found under ”FBD elements”. Please adhere strictly to the defined command sequence in order to avoid syntax errors. There can be additional instructions between commands.

Example: TI, pulse (monostable–circuit – monoflop)

```
:   A   I2.1        Start (input signal) timing element T30
:   SI  T30
:   DZB 100MS       Define the time base in ms: 10, 100, 1000, 10000
                    or 60000
                    or input 10MS, 100MS, 1000MS, 10000MS, 60000MS
:   L   K30         Load constant K30 for setpoint value
:   A   I2.2        Reset time block
:   R   T30
:   =   Q3.1        With 0 → 1 edge on I2.1: ”1” signal on Q3.1.
```

TV, extended pulse

```
:   A     I2.1
:   SV    T31
:   DZB   100MS
:   L     TSW31
:   A     I2.2
:   R     T31
:   =     Q3.1
:   ***
```

TE, on–delay

```
:   A     I2.1
:   SE    T32
:   DZB   100MS
:   L     TSW32
:   A     I2.2
:   R     T32
:   =     Q3.2
:   ***
```

TS, stored on–delay

```
:   A     I2.1
:   SS    T33
:   DZB   100MS
:   L     MW32
:   A     I2.2
:   R     T33
:   =     Q3.3
:   ***
```

TA, off–delay

```
:   A     I2.1
:   SA    T34
:   DZB   100MS
:   L     MW33
:   A     I2.2
:   R     T34
:   =     Q3.4
:   ***
```

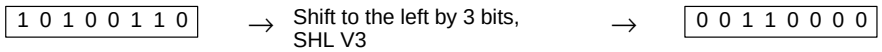
Edge recognition
The following operations are able to detect pulse edges.

FLP	Detection of an edge transition 0 → 1
FLN	Detection of an edge transition 1 → 0
FL	Detection of edge transitions 0 → 1 and 1 → 0

Examples

```
FLP                FLN                FL
:   A     I2.1     :   A     I2.1     :   A     I2.1
:   FLP   M1.10    :   FLN   M1.10    :   FL    M1.10
:   =     M1.11    :   =     M1.11    :   =     M1.11
:   ***            :   ***            :   ***
```

Shift (only in IL)
These operations can be used to shift the contents of the register by a user–defined number of bits. The moved bits are lost. Sign information would be lost. Therefore, shift is only sensible in logic.



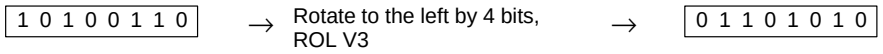
Example:
SHL, shift to the left

```
:   A     MB1     The contents of MB1 are moved 3 bits to the left.
:   SHL   V3      SHL is only possible with constants.
:   =     MB1     The moved value is in MB1.
:   ***
```

Example:
SHR, shift to the right

```
:   A     MB1     The contents of MB1 are moved 5 bits to the right.
:   SHR   V5      SHR is only possible with constants.
:   =     MB1     The moved value is in MB1.
:   ***
```

Rotate (only in IL)
These operations can be used to rotate the contents of the register by a user–defined number of bits. The rotated bits are reinserted in the opposite direction.



Example:
ROL, rotate to the left

```
:   A     MB1     The contents of MB1 are rotated 4 bits to the left.
:   ROL   V4      ROL is only possible with constants.
:   =     MB1     The moved value is in MB1.
:   ***
```

Example:
ROR, rotate to the right

```
:   A     MB1     The contents of MB1 are rotated 5 bits to the right.
:   ROR   V5      ROR is only possible with constants.
:   =     MB1     The moved value is in MB1.
:   ***
```

Incrementing/decrementing (only in IL)
The following functions can be used to increase or decrease the contents of a byte, word or doubleword by 1. This is used to modify setpoint values, for example.

INC increase value by 1
DEC decrease value by 1

Example

```
INC                     DEC
:   A     MW1           :   A     MB1
:   INC                 :   DEC
:   =     MW1           :   =     MB1
```

Load, Transfer (only in IL)

LBB, LBW: bit → byte, word
Operations LBB and LBW are used to load (read into the register) a defined bit string (byte register, word register).
The operand following the load operation is the least significant binary signal of the bit string. Upon input, an equipment list dependent syntax control of the acceptability of the bit string is carried out.

TBB, TBW: byte, word → bit
Operations TBB and TBW assign the register contents (binary) to a bit string. The transfer begins with the bit of least significant value from the register content. The start of the bit string is determined by the operands.

In order to avoid damage to memory areas and undesired memory content, a function is introduced in which the user states the width of the bit string.
The statement of bit string width is carried out with the command

DBB ANZ xx Define bit string width

The operand number (ANZ) can accept the following values:

LBB, TBB with bit operand
DBB ANZ 1...8 for maximum 8 bits in byte register
Formal parameter in FB declaration part: B8

LBW, TBW with bit operand
DBB ANZ 1...16 for maximum 16 bits in word register
Formal parameter in FB declaration part: B16

LBW, TBW with byte operand
DBB ANZ 1...2 for maximum 2 bytes in word
Formal parameter in FB declaration part: B2

Rules for creating IL:
It is absolutely necessary that the instruction DBB directly follows its bit string instruction. In the formal parameter declaration, the declaration for the number must directly follow the declaration of the address. Otherwise, testing and assignment are not possible. The declaration of the setpoint types B2, B8 and B16 must be followed by the declaration of a setpoint type ANZ.

☞ **Note** The maximum number of instructions and moved bits in a user function block (DBB ANZxxx) is $\leq$ 3.000.
Σ instructions + Σ xxx $\leq$ 3.000

Example:

```
Network 1 FB3
NAME:     FBTEST
<Identifier       Required type     Attribute>
IDENT:BYE1        B2                I
IDENT:BSP1        ANZ               I
IDENT:BYA1        B16               O
IDENT:BSP2        ANZ               O
IDENT:BIT2        B8                I
IDENT:BSP3        ANZ               I
IDENT:BIA2        B8                O
IDENT:BSP4        ANZ               O
IDENT:BIT3        B16               I
IDENT:BSP5        ANZ               I
IDENT:BIA3        B16               O
IDENT:BSP6        ANZ               O
***

Network 2
:   LBW   =BYE1          Load 1...2 bytes in word register
:   DBB   =BSP1          Define bit string width 1...2
:   A     KH000F
:   TBW   =BYA1          Transfer word register to bit string
:   DBB   =BSP2          Define bit string width 1...16
:   :
:   =     MW12
:   ***
```

```
Network 3
:   LBB   =BIT2      Load bit string in byte register
:   DBB   =BSP3      Define bit string width 1...8
:   A     MB10
:   TBB   =BIA2      Transfer byte register to bit string
:   DBB   =BSP4      Define bit string width 1...8
:   :
:   =     MB20
:   ***

Network 4
:   LBW   =BIT3      Load bit string in word register
:   DBB   =BSP5      Define bit string width 1...16
:   A     MW10
:   TBW   =BIA3      Transfer word register to bit string
:   DBB   =BSP6      Define bit string width 1...16
:   :
:   =     MW13
:   ***

Network 5
:   LBW   E8.12      Absolute addressing
:   DBB   =BSP5      Define bit string width 1...5, since I8.12...I8.16;
:   =     MW22       if LBW, LBB is absolute, ANZ xx must be
:   TE               entered.
```

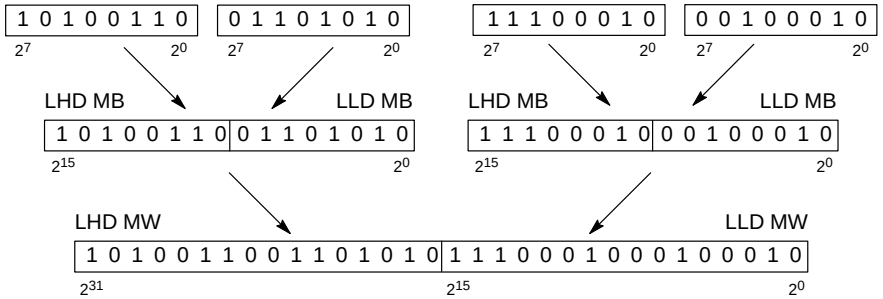
LLD, LHD: Load Low Data/ Load High Data → word, doubleword

The operations LLD and LHD load:
bytes from the signal memory into the word register, or
words from the signal memory into the doubleword register

A word consists of a high byte and a low byte. The low byte contains the less significant word part, the high byte contains the more significant word part. Accordingly, a double word consists of a high word and a low word.

With the command LLD the user can:
define any byte as the low byte of a word, or
define any word as the low word of a doubleword

With the command LHD the user can:
define any byte as the high byte of a word, or
define any word as the high word of a doubleword



Examples:

Using the operation LLD, the contents of MB112 (decimal 100) are loaded into the low word of marker word MW223.

```
:L      K0
:=      MB113                   00000000
:LLD    MB112                   01100100
:LHD    MB113                   00000000
:=      MW223       00000000    01100100
```

Using the operation LLD, the contents of MW212 (decimal 32 612) are loaded into the low word of marker double word MD200.

```
:L    V0
:=    MW213                              00000000 00000000
:LLD  MW212                               01111111 01100100
:LHD  MW213                              00000000 00000000
:=    MD200      00000000 00000000 01111111 01100100
```

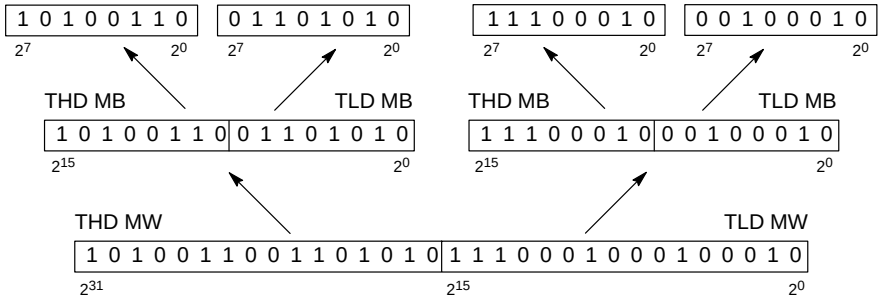
TLD, THD: Transfer Low Data/ Transfer High Data → word, doubleword

The operations TLD and THD transfer:
bytes from the word register into the signal memory, or
words from the doubleword register into the signal memory

A word consists of a high byte and a low byte. The low byte contains the less significant word part, the high byte contains the more significant word part. Accordingly, a double word consists of a high word and a low word.

With the command TLD, the user can:
define the low byte of a word as any byte, or
define the low word of a doubleword as any word

With the command THD, the user can:
define the high byte of a word as any byte, or
define the high word of a doubleword as any word



Examples

Using the operations TLD and THD, the contents of marker word MW219 (decimal 1 024) are transferred into low byte MB117 (0) and MB118 (4).

```
:L     MW219      00000100  00000000
:TLD   MB217                00000000
:THD   MB118                00000100
```

Using the operations TLD and THD, the contents of marker double word MD17 (decimal 2 097 168) are transferred into low word MW220 (16) and high word MW221 (32).

```
:L    MD17    00000000 00100000 00000000 00010000
:TLD  MW220                     00000000 00010000
:THD  MW221                     00000000 00100000
```

The analog value EW5.1 (15 928) is loaded into the marker double word MD18. Then the contents are loaded into the low word MW69 and the highword MW70.

```
:L    V0
:=    MW68                      00000000 00000000
:LLD  IW5.1                      00111110 00111000
:LHD  MW68                      00000000 00000000
:=    MD18    00000000 00000000 00111110 00111000
:L    MD18    00000000 00000000 00111110 00111000
:TLD  MW69                       00111110 00111000
:THD  MW70                      00000000 00000000
```

Indirect addressing (signal type pointer)

General

Pointers provide the user with the ability to call–up **all** the addresses in the installed memory volume of an ALU **one byte at a time** and also to **change them.**

Warning Since the system allows access to all signals including system signals, special care must be exercised when configuring with the pointer. This applies especially to program loops in which the address range and step width will be changed according to the function.

Note With effect from version 7, there is a selection point called ”Pointercheck” under the menu SETUP / PLC STATION. With Pointercheck enabled (yes), a check is carried out on each read/write procedure to see whether access is allowed. It should be kept in mind that this checking adds an overhead to program runtime. If the setting is changed, a new PLC Link and Load is required.

Warning When the signal memory ranges parameterized in the equipment list are being changed, monitoring by means of ”Pointercheck = Yes” should be enabled, at least for test purposes.

Note After changes to memory partitioning, the program must be linked again.

These freedoms have been made possible by the internal use of the **physical** address(es) on which every user signal is based. This is then known as **indirect** addressing.
Simply by increasing the contents of the pointer (**one byte at a time**), andthere-

by the physical address, each **byte** of **any** signal address can be accessed and **changed**.

This is also known as **indirect** addressing. User actions then consist of e.g. **calling** and **processing signal memory ranges** whose contents need to be changed because, for instance, they are **process dependent**.

Examples of activities connected with this could be:

Reloading address contents into another area (e.g. formulae)
Comparing address contents (searching tables/data structures for particular values)
Copying data strings such as bytes or words.
Pointers as parameters in FBs
Transferring parameters by FB to SFB
Processing indirectly addressed values

Further examples can be found in the Help texts.
(Call via <F10>, <F10>, ”IL(instruction list)”, ”Indirect addressing and examples”,”Examples”)

Additional information

Pointers apply to the whole program.

A pointer always **indicates** a **byte** address (physical address).
Using various load commands it is possible to load:

- the contents of the pointer,
- the address of a pointer or
- the contents of the address indicated by the pointer

(see ”Load” on page142).

Initial values for the data type address (in data structures) and pointer are permissible. An initial value is to be entered as a signal identifier in ASCII. (e.g. MW105, GEDB5.2, ...).

Addresses and pointers can be assigned using the initial value input in the SYM/COM editor or by assignment in the user program.

☞ **Note** Symbolic identifiers are not permissible for the data type address and pointer.

Pointers on SFBs are not possible at present. These will, however, be considered for future SFBs.

Indirect addressing of blocks (OB, PB, FB, AE, AZ, TB) and labels is not possible.

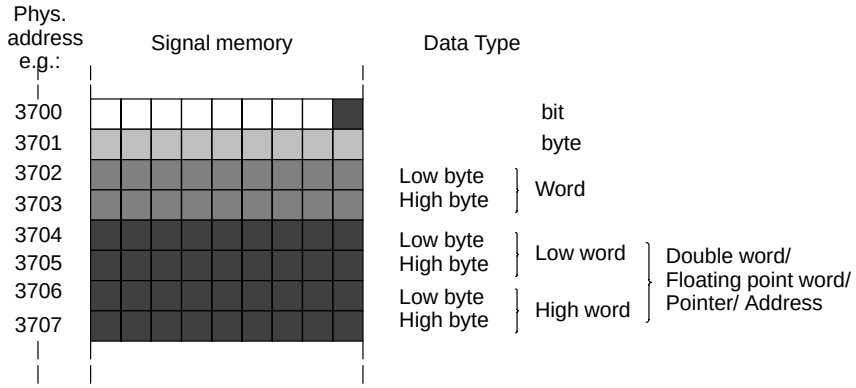
Addressing
A pointer always indicates a byte address. All data types (bit, byte, word, double word, floating point word, address and pointer) are addressed with byte adresses. Several data types (word, double word, etc.) occupy several successive byte addresses. The pointer always indicates the first address in the assigned ”signal memory area”. During pointer processing, the programmer must insure that the pointer is adjusted by the correct value, i.e. the width of the data type must be taken into account.

⚠ **Caution Incorrect pointer addressing can cause a program to crash if ”Pointercheck = No” has been specified.**

Pointer indicates data type bit or byte,
then next pointer address = first pointer address +1.

Pointer indicates data type word,
then next pointer address = first pointer address +2.

Pointer indicates data type double word, floating point word, pointer or address,
then next pointer address = first pointer address +4.



Pointer option

For indirect loading, the pointer is expanded with an option that sets the register to the correct data width (Pnnn–>BIT/ –>BYTE/ –>WORD/ –>DWORD/ –>FWORD/ –>Pointer). For input into IL, the ”arrow” consists of the characters ”–” and ”>”.

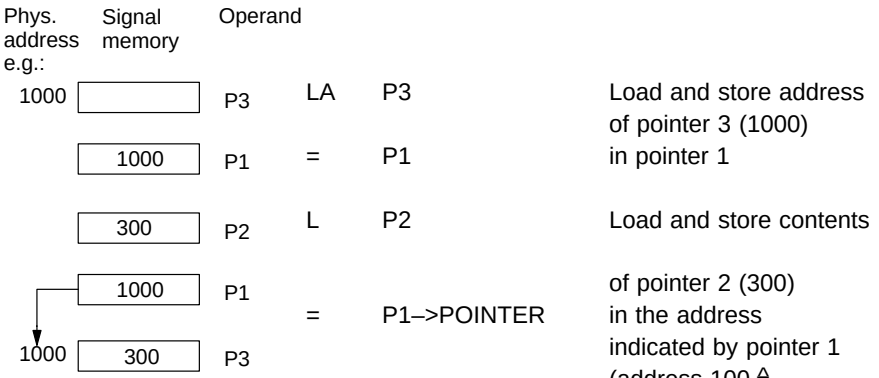
Permissible operations

The following operations are permissible in connection with the operand Pnnn:

- Load operations: L, LA
- Assignments: =, =:
- Arithmetic operations: ADD, SUB, INC, DEC
- Comparison operations: >, >=, ==, <, <=, <>

Null pointer
There is a null pointer P_NULL, which indicates address 0. It enables a comparison at address 0.

Example:
Pointers indirectly assign another pointer
The contents of pointer 2 shall be indirectly assigned (via pointer 1) to pointer 3.



Pointer 2 and pointer 3 now have the same contents.

An example for the practical application of this IL feature can be found in the user manual ”ALD for Beginners” chapter 8.10., page 140.

Further detailed examples of indirect addressing are also found in the user manual ”ALD for Beginners”.

Jumps (only in IL)

Inputting jump marks

When creating a network in IL, you can include conditional and unconditional jumps within it. The mark that is entered for the jump call must also be entered at the jump goal. Jump marks are introduced with an ”=” sign, just like formal parameters.

The jump goal should be a load instruction, in the case of arithmetic, or the start of a logical connection, since the register is thereby initialised.
Since the register is initialised after an assignment, this has an effect on the jump goal (see Example SPB, conditional jump)

Caution If an ”endless loop” has been configured, a program transfer to PLC is not possible. A renewed bootloading of the ALU is necessary and the program must be corrected and linked again.

Example: SP, unconditional jump

```
          :   A   I2.1        Jump mark goals are entered to the left of
          :   A   I2.2        the colon. You will need to move the cursor there
          :   =   Q3.1        to do this (arrow keys or mouse). Jump marks
          :   SP  =AM0012     consist of up to six characters. The first of
          :   ...             these must be a capital letter. The rest of
AM0012    :   A   Q3.1        the characters can be numbers, special characters
                               or
          :   ...             letters, but spaces are not allowed.
```

Example: SPB, conditional jump with initialization

```
          :   S     I2.3
          :   A     I2.4
          :   SPB   =AM0022     The jump is carried out
          :   ...               on a ”1” signal.
          :   =     M4.1
AM0022    :   A     I2.5        with initialization
          :   ...
AM0022    :   =     Q3.2        without initialization
```


Note Reproduction of the jumps in LD and FBD is not available.

NOP instruction (only in IL)
A ”NOP” instruction is a dummy operation.

☞ **Note** ILs with ”NOP” instructions cannot be reproduced in LD or FBD.

FREE instruction (only in IL)
FREE is a dummy in instruction list. Unlike NOP, which can stand for any instruction, FREE is only valid for inputs from memories, timers and counters.

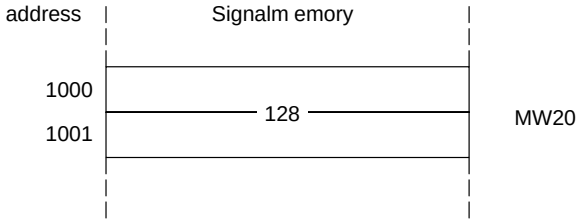
Example:

without dummy			with dummy			
:	A	I2.1	:	A	I2.1	In this network, note that
:	SV	T31	:	SV	T31	the time block cannot be
:	DZB	100MS	:	DZB	100MS	reset when it is
:	L	V10	:	L	V10	with dummy.
:	A	I.2	:	FREE		
:	R	T31	:	=	Q3.1	
:	=	Q3.1	:	***		
:	***					

Arithmetic functions
Load
Arithmetic functions always start with ”L”.

Loading operand addresses
To load operand addresses, the load command LA (load address) is used.

Differences between the load commands



Catchword MW20 occupies addresses 1000 and 1001 in the signal memory. The contents of MW20 are 128.
The IL extracts which follow show the various load commands by way of example.

LA	MW20	Load address from MW20. The address (1000) of the first byte occupied by MW20 is loaded into the register.
= (P1)	P1	The contents of the register (1000) are loaded into pointer 1
L	P1	Load the contents of P1. The contents (1000) of pointer 1 are loaded.
L tents	P1–>WORD	Indirectly load the contents of P1 in word register. The con- of addresses 1000 and 1001 are loaded (128). Through pointer option ”–>WORD”, the pointer is informed that the information has word width.

Constants
Constants V, VO and VH are loaded with operation ”L”.

Calculating

Essentially all arithmetic operations must begin with a load operation.

Example

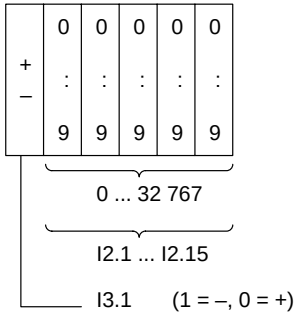
```
:   L     K12
:   ADD   MW1      Several arithmetic commands can follow
:   SUB   MW12     one another
:   =     MW20
:   ***

:   L     MW22     In arithmetic, there is no no hierachy of operations.
:   MUL   MW33     The expression <MW11> + <MW22> x <MW33> =
:   ADD   MW11     <MW44> must be programmed as shown
:   =     MW44     here.
:   ***
```

It is possible to convert between data types, see ”Converting”.

If system markers are to be used, they must be called directly after the arithmetic operation.

Generating two's complement

For BCN switches with a separate sign, the two's complement should be calculated for negative values.



corresponding IL:

```
      :   LBW   I2.1
      :   DBB   ANZ
      :   =     MW1
      :   A     I3.1
      :   SPZ   =END
      :   L     K 0
      :   SUB   MW1
      :   =     MW1
END   :   L     MW1
      :   =     MD1
```

Comparator

Comparators can only be used with operands of the same data type (in accordance with conversion rules).

If you do not require reproduction in LD/FBD, arithmetic instructions are permissible between load and comparison operations. Several instructions on bit logic operation can precede the ”=” assignment.

= equal to

```
:   L     MW30
:   = =   MW31
:   =     Q3.1
:   ***
```

> greater than

```
:   L     MW32
:   >     MW33
:   =     M5.22
:   ***
```

< less than

```
:   L     MW50
:   <     MW51
:   =     Q3.3
:   ***
```

<> does not equal

```
:   L     MW52
:   < >   MW53
:   =     Q3.4
:   ***
```

≥ greater than/equal to

```
:   L     MW40
:   >=    MW42
:   =     M5.23
:   ***
```

≤ less than/equal to

```
:   L     MW
:   <=    MW52
:   =     Q3.6
:   ***
```

Conversion

☞ **Note** There are two types of processing and various types of operand. The possible conversions occur automatically. The user can use mixed operand types and mixed types of processing in IL. The following functional description makes clear how the values in the dynamic display arise.

Converting into other operand types

- Logic: Operand types without sign
  bit, byte, word and double word
  including e.g. the following operations: A, O, X, AN

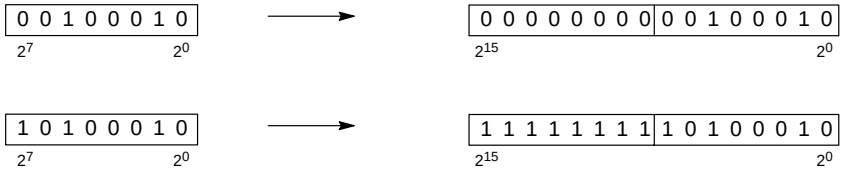
- Arithmetic: Operand types with sign
  byte, word and double word
  including e.g. the following operations: ADD, MUL, DIV, SUB

- The following operations do not influence the conversion:
  SHL, SHR, LBS, ROR, ROL, SIG, DEC, INC

Conversion is always from a smaller to a larger operand type.

- Logic: Operand type without sign; when converting, the more significant unassigned bits are filled with ”0”.
  Example: Converting from MB to MW

- Arithmetic: Operand types with sign; when converting, the more significant unassigned bits are filled according to sign.
  More significant bits with positive values are filled with ”0”.
  More significant bits with negative values are filled with ”1”.
  Examples: Converting from MB to MW.



Conversion between two types of processing (logic → arithmetic, or arithmetic → logic)

- The conversion from logic to arithmetic is always without sign.

- The conversion from arithmetic to logic is always with sign.

Exponential functions/sign handling

The exponential functions can generally only be carried out with floating point word operands (e.g. MG..., VG...).

Examples:

IL:		Contents	Description
L	VG4E0	4	The constant is loaded as base in the register
EXPT	MG1	5	. The content of MG1 is used as
=	MG2	1024	exponent. The power is stored in MG2. (4^5=1024)
L	VG12E0	12	The constant is loaded and as exponent
EXP			the base e is used. The power is stored in
=	MG3	162755	MG3. (e^{12}=162755)
L	MG20	5	The square root of the contents
SQRT			of the MG20 is taken and stored in MG4.
=	MG4	2.23607	($\sqrt{5}$ = 2.23607)
L	KG7E0	7	The reciprocal of the constant is taken
1/X			and the result is stored in MG5.
=	MG5	0.142857	($^1/_7$=0.142857)
L	VG7E0	7	The natural logarithm of the
LN			constant is taken and the result is
=	MG6	1.94591	stored in MG6. (ln 7=1.94591)
L	VG9E0	9	The logarithm to the base 10 of the
LOG			constant is taken and stored in MG7.
=	MG7	0.954243	(log 9=0.0954243)
L	VG–5E0	–5	The absolute value of the constant
ABS			is taken and stored in MG8.
=	MG8	5	(I–5I=5)

IL:		Contents	Description
L	MG19	5	The sign of the contents of
SIG			MG19 is determined and stored in M1.1
=	M1.1	1	(1 = ”0” or positive sign;
			0 = negative sign)

Modulo function

The modulo function is carried out with byte, word or double word operands.

Examples:

IL:		Contents	Description
L	MW100	100	The content of MW100 (divisor) is divided by
MOD	MW80	80	the content of MW80 (dividend) and
=	MW5	20	the remainder (modulo) is stored in MW5.

Angle functions

The angle functions can generally only be carried out with floating point word operands (e.g. MG..., VG...). Angles are to be stated in radian measure in ALD25.

Examples:

IL:		Contents	Description
L	VG4E0	4	The sine of the constant (in
SIN			radians) is calculated and the
=	MG10	–0.756802	result is stored in MG10.
L	VG12E0	12	The cosine of the constant (in
COS			radians) is calculated and the
=	MG11	0.843854	result is stored in MG11.
L	MG22	15	The tangent of the constant (in
TAN			radians) is calculated and the
=	MG12	–0.855993	result is stored in MG12.

IL:		Contents	Description
L ASIN =	VG0.5E0 MG13	0.5 0.523599	The arc sine of the constant (in radians) is calculated and the result is stored in MG13.
L ACOS =	VG0.5E0 MG14	0.5 1.0472	The arc cosine of the constant (in radians) is calculated and the result is stored in MG14.
L ATAN =	MG23 MG15	0.5 0.463648	The arc tangent of the contents of MG23 (in radians) is calculated and the result is stored in MG15.

Ladder diagram LD

OBs, PBs , FBs, AEs, AZs and TBs can be programmed in LD.
Per ladder diagram network, as inputs, a maximum of 16 signals in parallel and 7 signals plus output in series can be linked. There is only one output per LD network. This cannot be negated. The valence of the output signal can be set on a maximum of 16 contacts.
There are no jumps in LD, instead only block calls.
Processing of the called block is followed by return to behind the place that called.
In LD, a program or function block can only be alone in a network.
When generating networks in LD, one must be aware that the elements can only be entered in certain cursor positions.

The following symbols can be selected when generating networks in LD:

	normally open contact (inquiry 1 signal)
	normally closed contact (inquiry 0 signal)
	connector
=2k+1	exclusive OR block
	connection of parallel paths continuation in parallel path without contacts
	output (not stored)
FBD elements	further elements that can be called in LD and FBD (memories, timers, counters and comparators) can be selected and called.

☞ **Note** on FBD elements: Normally, when programming in LD, only one FBD element per network is possible. Exceptions are several timers, counters, memories beside one another in the upper cross–path.

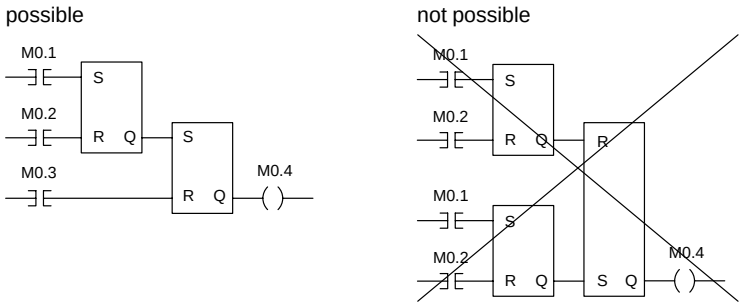


Figure 1 Inputting FBD elements in ladder diagram

Function block diagram FBD

OBs, PBs, FBs , AEs, AZs and TBs can be programmed in FBD.
Function block diagram allows a maximum of 6 FBD elements per network horizontally, and 46 input signals together with a maximum of 12 FBD elements with two inputs each vertically. There is only one output per FBD network. This cannot be negated.
The valence of the output signal can be set on a maximum of 16 contacts .

In FBD there are no jumps, instead only block calls.
Processing of the called block is followed by return to the place from which the call was made.
In FBD, a program or function block call must always be alone in a network.

For generating networks in FBD, it is important to note that elements can only be entered at certain cursor positions.

The following symbols can be used for geneating networks in FBD:

&	AND block
>=1	OR block
=2k+1	exclusive OR block
——|	input (inquiry 1 signal)
——o|	negated input (inquiry 0 signal)
—> >—	connector
FBD elements:	The elements that can be called in LD and FBD (memories, timers, counters and comparators) can be selected and called.

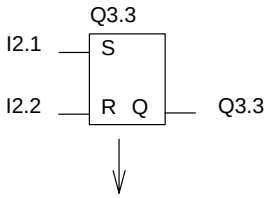
FBD elements
FBD elements are standard blocks that you can select in the IL, FBD or LD Editor.

All AKF functions (FBD elements) show latching properties, i.e. following momentary power drop–out on the PLC all signals and words are unchanged. To achieve initial state characteristics (Initialisation upon voltage recovery), the system marker SM39 should be used as the Initialisation input of the function.

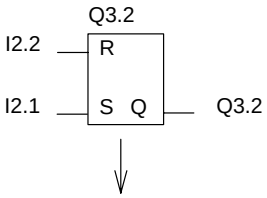
An example of each FBD element is shown below.

Memory

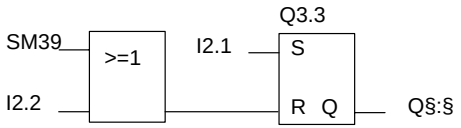
SR Flipflops
dominant reset



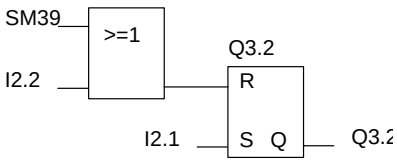
RS Flipflops
dominant set



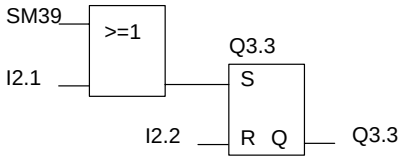
preferred state after power–on 0 (initial state)



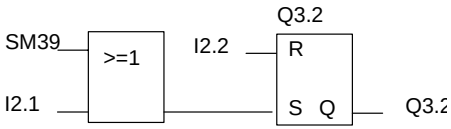
preferred state after power–on 0 (initial



preferred state after power–on 1



preferred state after power–on 1



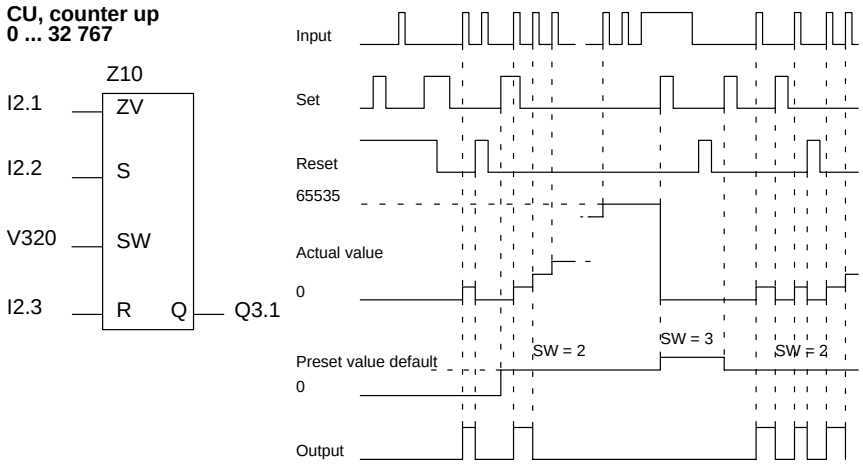


Figure 2 FBD element and timing diagram counter up

With a ”1” signal on reset input ”R”, the actual value and the output ”Q” are set to ”0”.

With a 0 → 1 edge on set input ”S”, the setpoint value ”SW” is imported into the set point value register. The actual value becomes ”0”. Only after at least one setpoint value has been accepted, is a setpoint value/actual value comparison undertaken.

The permissible setpoint values ”SW” are: K, MW, ZSW

The output ”Q” is set to ”1” when the actual value is between 0 and the setpoint value.

With a 0 → 1 edge on input ”CU”, the actual value is increased by 1 (up to a maximum of 65 535) and compared with the setpoint value.
The actual and setpoint values (ZIW, ZSW) in the counter can be controlled through the on–line list and/or initial values. The actual and setpoint counter values can be scanned using a load assignment and transferred to other operand areas.

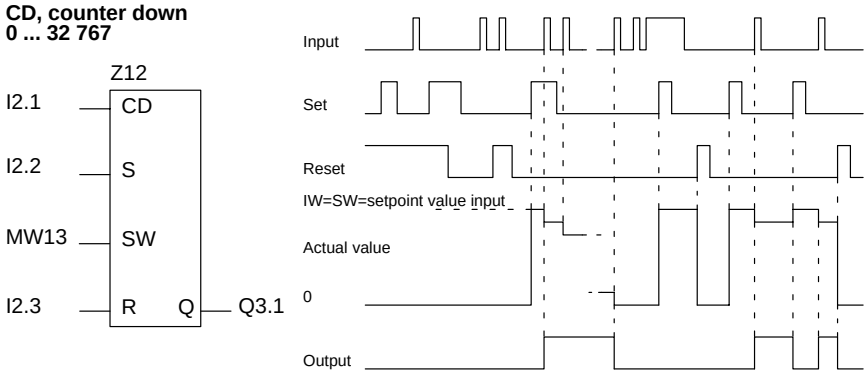


Figure 3 FBD element and timing diagram counter down

With a ”1” signal on reset input ”R”, the actual value and the output ”Q” are set to ”0”.

With a 0 → 1 edge on set input ”S”, the setpoint value ”SW” is imported into the set point value register. The actual value is set equal to the setpoint value. Only after at least one setpoint value has been accepted, is a setpoint value/actual value comparison undertaken.

The permissible setpoint values ”SW” are: V, MW, CSW

The output ”Q” is set to ”1” when the actual value is between 0 and the setpoint value.

With a 0 → 1 edge on input ”CD”, the actual value is reduced by 1 (down to a minimum of 0) and compared with the setpoint value.

The actual and setpoint values (CIW, CSW) in the counter can be controlled through the on–line list and/or initial values. The actual and setpoint counter values can be scanned using a load assignment and transferred to other operand areas.

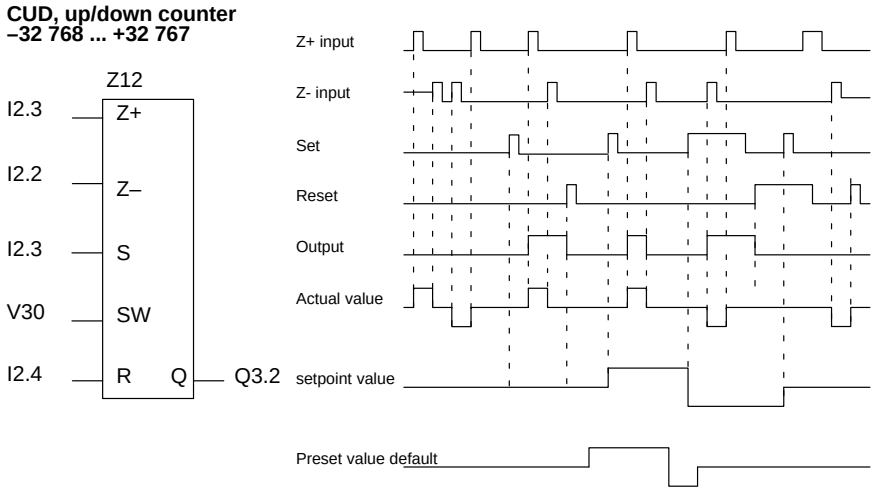


Figure 4 FBD element and timing diagram counter up, down

With a ”1” signal on reset input ”R”, the actual value and the output ”Q” are set to ”0”.

With a 0 → 1 edge on set input ”S”, the setpoint value ”SW” is imported into the set point value register. The actual value remains unchanged. Only after at least one setpoint value has been accepted, is a preset value/actual value comparison undertaken.

The permissible setpoint values ”SW” are: V, MW, ZSW

The output ”Q” is set to ”1” as soon as the actual value is greater than or equal to the setpoint value.

If the next count operation would result in the actual value becoming greater than 32 767 or less than –32 768, the system marker ”actual value overflow” (SM 638) is set to ”1”.
The system marker is automatically reset before the counter is called again.

With a 0 → 1 edge on input Z+, the actual value is increased by 1.
With a 0 → 1 edge on input Z–, the actual value is reduced by 1.

The actual and setpoint values (ZIW, ZSW) in the counter can be controlled through the on–line list and/or initial values. The actual and setpoint counter values can be scanned using a load assignment and transferred to other operand areas.

Timers

TI, timer function pulse

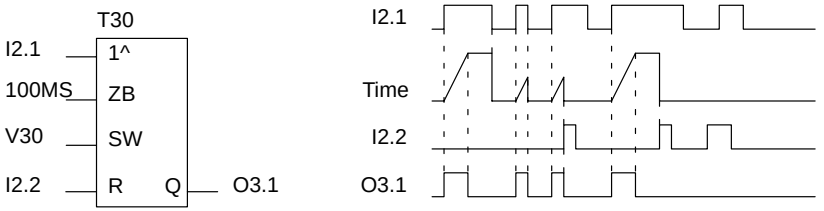


Figure 5 FBD element and timing diagram pulse

TV, timer function extended pulse

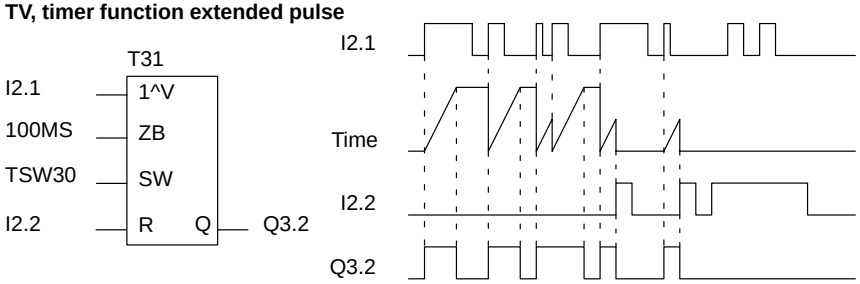


Figure 6 FBD element and timing diagram extended pulse

TE, timer function on–delay

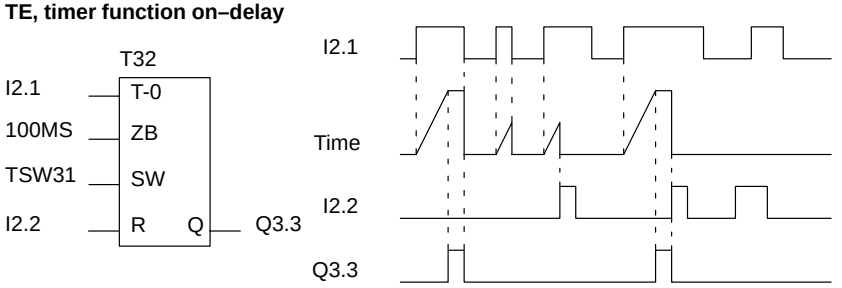
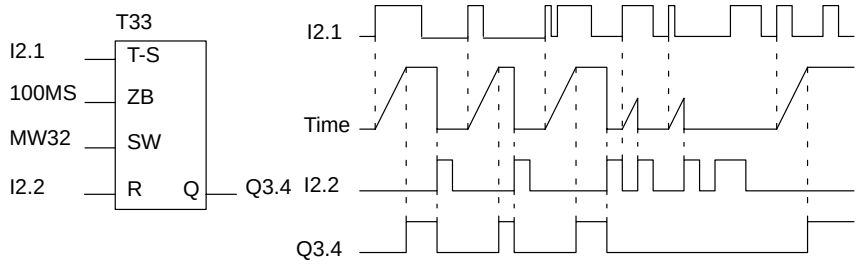


Figure 7 FBD element and timing diagram on–delay

TS, timer function stored on–delay



reset only possible with reset

Figure 8 FBD element and timing diagram stored on–delay

TA, timer function off–delay

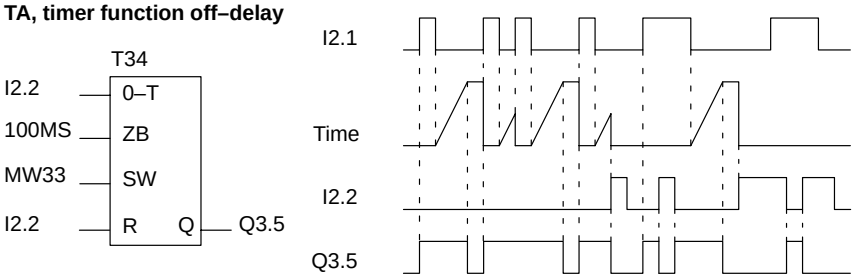
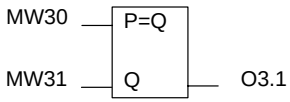


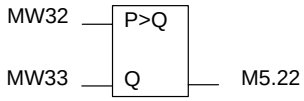
Figure 9 FBD element and timing diagram off–delay

Comparator
The contents of two words are compared. Six different comparators can be used.

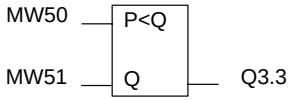
= **equal to**



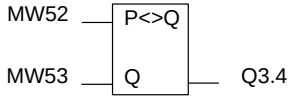
> **greater−than**



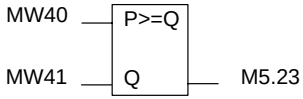
< **less−than**



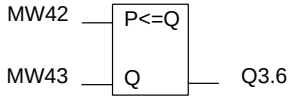
<> **not equal to**



>= **greater than/equal to**



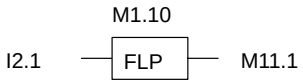
<= **less than/equal to**



When the condition described in the FBD element is met, output ”Q” = 1.

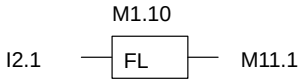
Edge recognition
Three FBD elements are available for pulse edge detection.

FLP, Edge detection 0 –> 1



FLN, Edge detection 1 –> 0

FL, Edge detection 0 –>1 and 1 –> 0



When the condition described in the FBD element is met, the output for a cycle ”Q” = 1.

The intermediate marker (here M1.10) and the output marker (here M11.1) cannot be identical.

During network generation in FBD, the syntax is checked by the Dolog ALD Software and invalid inputs are rejected.

Additional information on network creation in function block diagram can be found in the FBD editor and in the ”correction mode” of the FBD editor.

3.3.3 Symbols and Comments

-”Edit”, ”Symbols and Comments”

This software function serves to generate a SYM/COM block. It contains symbolic names, comments, and initial values for signal addresses.

Press <Return> to access the SYM/COM block editor.

SYM/COM block
To clarify the connection between an absolute address (I/Os, markers, etc.) and its technological function, it is possible to provide absolute addresses with symbolic names and comments. In addition, signal addresses can be given initial values. Symbols, comments and initial values are stored in the SYM/COM block (databank) under the current station name.

The text of the symbolic names, comments, initial values and extended comments is stored in the SYM/COM block under the current station name.

After activating the symbols by setting the addressing to ”SYM”, the symbolic names entered in this editor can be used during programming as an alternative to the absolute addresses.

The SYM/COM block can be documented under ”Print”, ”Symbols and Comments”.

There are standard presettings for the system markers.

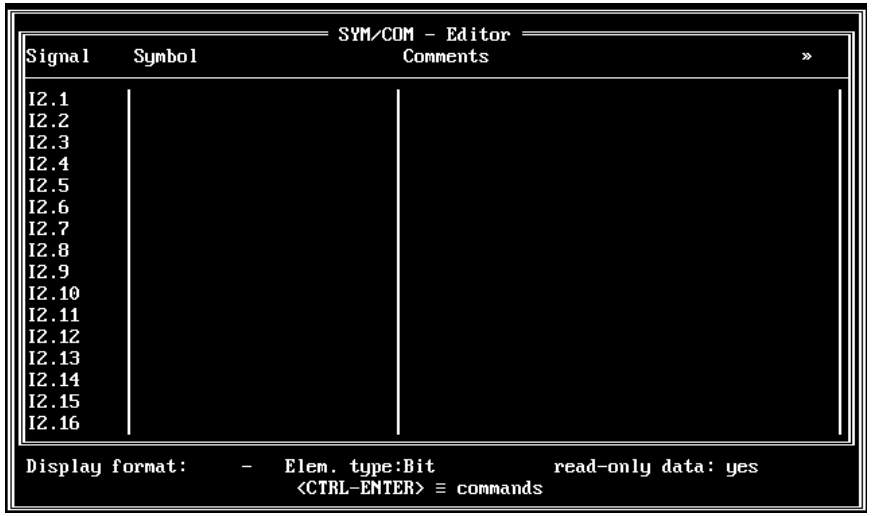
SYM/COM editor
This editor provides you with menus, macro keys, and a line editor. Within the editor you can use <Ctrl>+<Return> to call further menus. You can bypass the pulldown menu and call functions by pressing <Ctrl>+<reference character>.

Caution In order to assign a symbol to another address, you must select the function ”Replacement of Signals” (e.g. ANNA is to be assigned to pin I3.5 instead of I2.1), since this cannot be done here by amending. The program and PLC always work with absolute addresses. When ”SYM” is selected, the symbol is only displayed for the viewer.

Input Symbols and Comments

- ”Edit”, ”Symbols and Comments”, <Return>

Symbols are entered using the SYM/COM editor:
When the editor is started, a table showing the signal, symbol, initial value and comments for the first input is displayed.



You can use special keys, the line editor and macro keys to assign symbols, initial values and comments to the addresses shown.

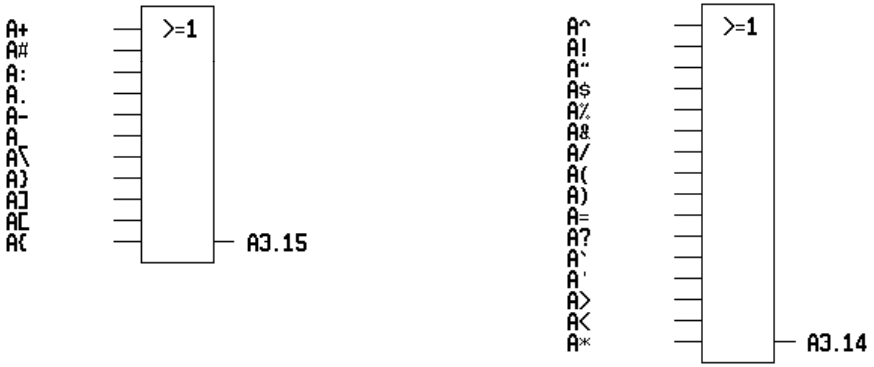
Symbols
A maximum of 24 characters can be entered for symbols.

☞ **Note** With COMAKF configuring you can use the first 8 characters only.

⚠ **Caution If the ALD25 or AKF125 station is to be visualized with Viewstar, symbol names must be specified for all signals used (maximum 24 characters).**

☞ **Note** Symbolic identifiers are not permissible for the data type address and pointer.

The following characters are possible:
0 to 9
A to Z, ÄÖÜß
the following special characters (the first character should be a letter, in this case ”A”)



☞ **Note** Symbol names which are also signal names are not allowed e.g. I2.1, M4.6

Comments
There are two different comments. The first comment consists of 40 characters. If you press <Ctrl>+<→> you can change to initial values and extended comments. The extended comment consists of up to 60 characters.

The arrow directions possible at the moment are shown in the upper right–hand corner of the screen. << means moving only with <Ctrl>+<←>, << >> means <Ctrl>+<←> or <Ctrl>+<→>, >> means only <Ctrl>+<→>.

Initial values
Press <Ctrl>+<→> to change to initial values and extended comments. Symbols or absolute addresses can be assigned an initial value. The function ”Program to PLC” loads values as blocks (initial value blocks IB) into the PLC. Initial starts and hot restarts will activate these blocks, thereby setting the signal memory and data structures. Initial values apply during the first scan on initial starts, cold restarts and hot restarts.

☞ **Note** Initial values are permissible for the data type address (in data structures) and for the pointer. An initial value is to be entered in ASCII foemat as a signal identifier. (e.g. MW105, GEDB5.2, etc.).

Please notice that the storage of initial values in the case of the Character display format (CHR) is **not** rotated in Intel format.

Example:

A word in display format:	**HEX**	**CHR**
Initial value input in the symbol and comment editor:	4142	AB
Storage in memory:	4241	4142

Special keys:

< ←>, < →>, < ↑>, <↓ >	Move the cursor
<Ctrl>+<←>, <Ctrl>+<→>	Move screen to left or right
<Del>	Erase character beneath cursor
<backspace>,< ⇐ >	Erase character to left
(tab)	Next input field
shift (tab)	Previous input field
<PgUp>	Previous page
<PgDn>	Next page
<Ctrl>+<PgUp>	Move line up
<Ctrl>+<PgDn>	Move line down
<Home>	Jump to top of screen
<End>	Jump to foot of screen
<Ins>	Insert / Overwrite
<Esc>	Terminate without saving
<Return>	Close off an input line
<F1>-<F8>	Call up text memory

Only those I/O addresses defined in the equipment list under ”number” are shown.

☞ **Note** All changes made with the functions of this editor apply only to the SYM/COM block and not to OB, PB, FB.

Press <Ctrl>+<Return> to call up a window with the following functions:

SYM/COM – Editor

Terminate (save)
Search Function
Search Operand
Erase Field/Line
Copy area
EraSe Area
Move Area
Copy Modified
SEarch Text
Replace Text
Search/Replace Again
Insert Linecomment
Generate Symbols

You can bypass the menu and call functions by pressing <Ctrl>+<reference character>.

3.3.3.1 Search Function

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>, ”Search Function”

You can search for either a signal or a symbol. After having chosen this function you will see a window appear on the screen in which the search termcan be entered. The input is checked for errors. You can terminate with either the <Esc> key or by entering a space. A search term that is found will be placed in the first line of the editor. An unsuccessful search will be followed by the corresponding message, and the cursor will remain in its original position.

For more information on this subject please see ”Search Operands” and ”Search Text”.

Search Function

Signals and Symbols can be searched.
Search for :

Please enter the exact signal designation, e.g. M1.1, I2.18, MW1330.

3.3.3.2 Search Operand

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>,
  ”Search Operand”

All operands (signal memory, data structure block) will be shown in one window.

To select an operand, use the arrow keys to move through the list and press <Return> to select, or use the mouse.

Enter the first possible signal/element of the chosen operand in the first line on the screen.

For more information please see ”Search Function”.

3.3.3.3 Erase Field / Line

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Erase Field / Line”

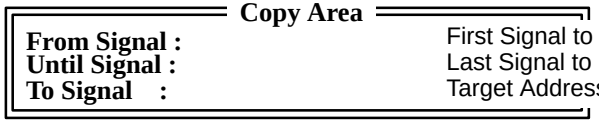
The input field on which the cursor was positioned before the menu was called up will be erase.

For more information please see ”Erase Area”

3.3.3.4 Copy Area

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Copy Area”

It is also possible to assign comments and initial values to other signals. The source of a copied comment is not erased.
Initial values will only be copied when the element type is the same.
After this function has been chosen, a window will appear on the screen.



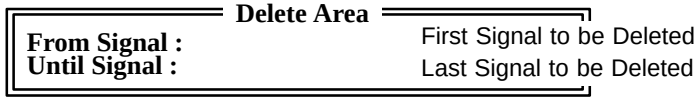
Input can be absolute as well as symbolic. The target address must be entered as an absolute. Overlapping ranges cannot be copied. A range can include a maximum of 200 samples.

Note No line comments will be copied, deleted, or relocated. In order to prevent double occupancy, no symbols will be copied here. The function ”Move Area” is used to assign new addresses to symbols.

3.3.3.5 Erase Area

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Erase Area”

It is possible to erase symbols, initial values and comments from a block.
After this function has been selected, a window will appear on the screen.



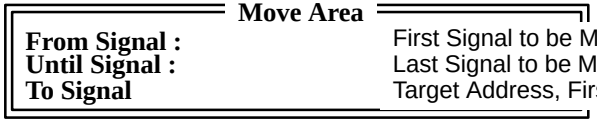
The entries can be either absolute or symbolic (target entry must be absolute).

Caution If modules are to be erased from the equipment list, all the symbols and comments concerned must first be deleted from the SYM/COM file.

3.3.3.6 Move Area

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Move Area”

It is possible to relocate symbols, initial values and comments as a block. Initial values will only be relocated if the element types are the same.
After this function has been chosen, a window will appear on the screen.



The entries can be either absolute or symbolic (target entry must be absolute). Overlapping ranges cannot be relocated. A range can include a maximum of 200 elements. After this function has been carried out, the texts and initial values will have disappeared from their original location (unlike Copy).

3.3.3.7 Copy modified

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Copy modified”

It is possible to assign symbols, comments and initial values by copying several target ranges. Initial values will only be copied if the element types are the same.

After this function has been selected, a window appears on the screen.

Symbols can only be copied once they have been changed. For this purpose, the following possibilities are available.

If search text and replacement text are specified, the part of the symbol specified by the search text will be replaced by the replacement text in the destination area.

If only a search text is indicated, the part in the destination area will be erased.

If no search text is indicated but a step width has been specified, a number separated by an underscore will be appended to the symbol in the target range.

The step width comprises a range of values from –100 to +100 or ’E’ for sample number.

If the specified search text contains a number, this number, depending on the step width, will be either incremented, decremented or replaced by the sample number in the destination area if no replacement text has been specified.

If neither search text nor replacement text and step width are specified, copying will take place without symbols.

If duplicate symbols result from copying, only comments and initial values (if data types are the same) will be stored in the destination area.

The input for the signals can be absolute or symbolic. If possible, the target address should be absolute.

The source area must be located within a sample number of a data structure. (e.g. M1.1 to M1.32, but M1.16 to M2.16 ist not possible).

An area can include a maximum of 200 elements.

Destination area
The destination area can be established by a target address, a sample list, or both.

Example:	from	to	to	Sample List	Destination Area
	M1.1	M1.5	M3.7	(none)	M3.7 to M3.11
	M1.1	M1.5	M3.7	4.7-9	M3.7 to M3.11 M4.7 to M4.11 M7.7 to M7.11 M8.7 to M8.11 M9.7 to M9.11

Example:	from	to	to	Sample List	Destination Area
	M1.1	M1.5		4.7-9	M4.1 to M4.5 M7.1 to M7.5 M8.1 to M8.5 M9.1 to M9.5

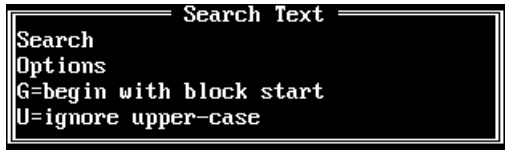
The destination area may not exceed the sample number.

Example:	from	to	to	Sample List	Destination Area
	M1.1	M1.5	M1.29		M1.29 to M2.1 is not possible.

3.3.3.8 Search Text

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Search Text”

It is possible to search for words, parts of words or parts of sentences among the comments in the SYM/COM block.
After this function has been selected, a window will appear on the screen.



”G” and/or ”U” can be entered under Options. You can bypass the pulldown menu and repeat the function as often as you like by using <Ctrl>+<W>.

Note To search for symbols and signal addresses, use the ”Search Function”.

3.3.3.9 Replace Text

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Replace Text”

This function allows you to search for any text in the SYM/COM block and replace it with a new text. The texts do not have to be the same length. In accordance with the search function, only comment texts can be searched.
After this function has been chosen, a window will appear on the screen.

”G” and/or ”U” and/or ”N” can be entered under Options. If ”N” is entered, the text will be replaced without you having to confirm. You can bypass the pulldown menu and repeat the function as often as you like by using <Ctrl>+<W>.

3.3.3.10 Search/Replace Again

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Search/Replace Again”

The parameters specified under ”Replace Text” can be executed as often as you wish with this function.

3.3.3.11 Insert comment line

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Insert Linecomment”

This function is used to insert a comment line above the present line. The present line will be pushed down. Further commentlinea can be inserted by repeating this function. A maximum of 62 characters per comment line can be entered.

3.3.3.12 Generate Symbols

- ”Edit”, ”Symbols and Comments”, <Ctrl>+<Return>
  ”Generate Symbols”

Symbols can be generated for an entire data structure, if comments have been allocated to the individual elements when they were defined in the data structure editor.

Part–symbols with a maximum of 15 characters can be entered for the individual samples in a window. The comment, now in capital letters, from the definition of the data structure which was allocated in the data structure editor will be attached to this part–symbol and, if possible, entered as a symbol.

The input of symbol names is closed with <Ctrl>+<B> and once confirmation is given, symbol generation starts. Input can be broken off by pressing <Esc>.

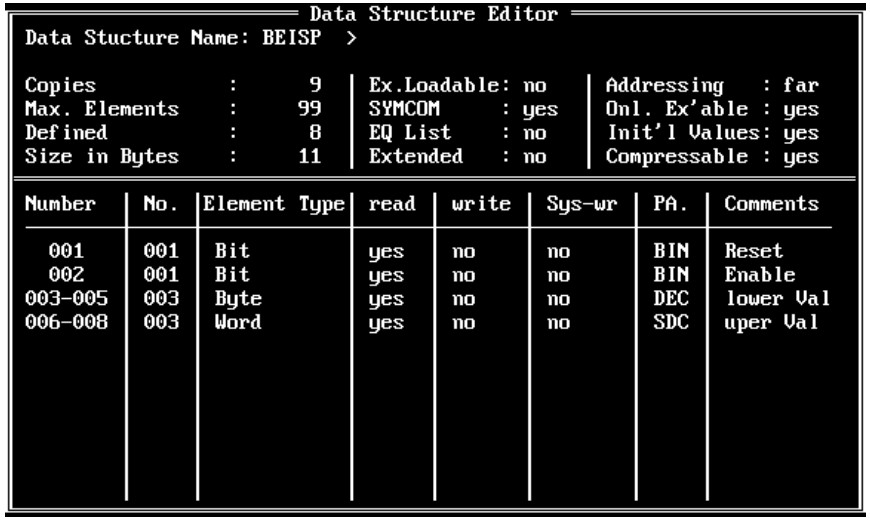
The spaces, commas and semicolons in the comment will be changed to underscores.

If a comment for several elements has been entered into the definition, a symbol will only be generated for the first element in the group.

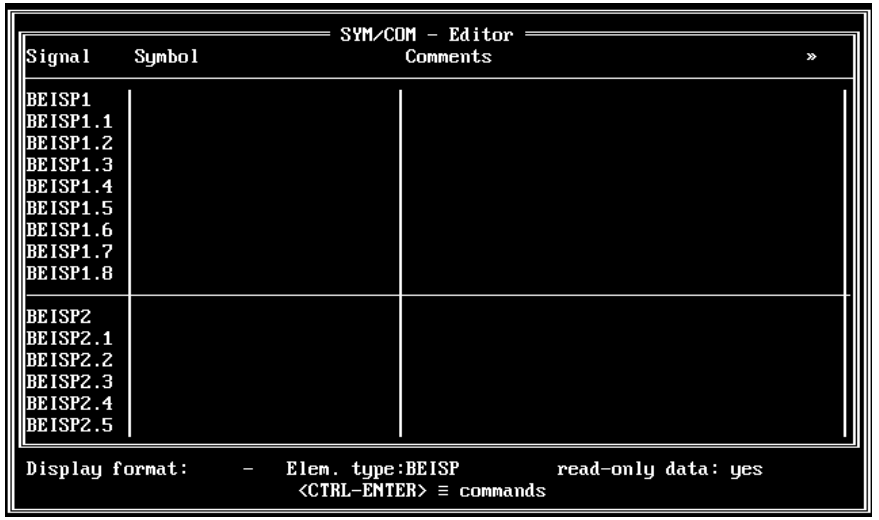
If symbols already exist for the data structure, a message will appear.

Example

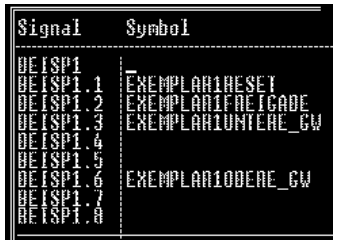
Organisation of the Data Structure (Data Structure Editor)


Data Structure Editor
Data Stucture Name: BEISP >
Copies : 9
Max. Elements : 99
Defined : 8
Size in Bytes : 11
Ex.Loadable: no
SYMCOM : yes
EQ List : no
Extended : no
Addressing : far
Onl. Ex'able : yes
Init'l Values: yes
Compressable : yes
Number | No. | Element Type | read | write | Sys-wr | PA. | Comments
001 | 001 | Bit | yes | no | no | BIN | Reset
002 | 001 | Bit | yes | no | no | BIN | Enable
003-005 | 003 | Byte | yes | no | no | DEC | lower Val
006-008 | 003 | Word | yes | no | no | SDC | uper Val

Input of Symbol Names (SYM/COM Editor)



Generated Symbols of the Data Structure Elements



Signal	Symbol
BEISP1	_
BEISP1.1	EXEMPLAR1RESET
BEISP1.2	EXEMPLAR1FAEIGABE
BEISP1.3	EXEMPLAR1UNTERE_GW
BEISP1.4	
BEISP1.5	
BEISP1.6	EXEMPLAR1ODERE_GW
BEISP1.7	
BEISP1.8	

3.3.4 Equipment list

- ”Edit”, ”Equipment List”

The equipment list contains the hardware equipment of the PLC to be configured.

Incompatibility of the Equipment List
An error can occur when the equipment list is being read at the start of the program (case 1)
or when it is being read by an ASCII import (case 2).

This can have the following causes:

- Possible Causes of Case 1
  - Technological developments lead to modifications or adaptations which invalidate the equipment list entries in question.
  - The user has made unauthorised changes in EAKARTEN.STD.

- Possible Causes of Case 2
  - Error during creation of the ASCII equipment List.
  - Import of an outdated equipment list.

In **case 1** the equipment list will be cleansed as far as possible by the system, that is, faulty modules will be deleted. For this reason you will only receive one error report. The next time ALD is called up, the cleansed equipment list will be used.

Errors can be corrected in the equipment editor (the automatic equipment list cleansing may already be to your liking).

If you receive an error report you should always call up the equipment editor, check for faulty modules and quit the equipment editor (Save a Copy) in order to ensure complete consistency of data.

With the help of an ASCII editor you can also process the ASCII equipment list ”_BES25.ABL” created by the system (see case 2).

In **case 2** no import is carried out, i.e, the equipment list in use before the import remains unchanged.

Answer the question, ”Display error analysis? (yes/no)” with ”yes” and make a note of the lines and fields where there have been errors in your equipment list (or see error file ”*.FBL”). Then use an ASCII editor to amend the import file you created.

Locate the lines concerned and correct the entries. Compare if neccessary the data here with that in EAKARTEN.STD. Import the corrected files into the equipment list.

System changes that lead to this error message:

If an equipment list was created with AKF V3.0x containing a TXT102 or a POS102/POS112, this creates an error report with effect from ALD V4.1. In the case of POS a new module will automatically be entered, and you will have to reparameterize the job element ”PIS” in the equipment editor. All TXT modules must be fully re–entered and reparameterized.

Equipment list for InterBus module interface
To couple the A250 to the remote bus, enter the BKF 102 remote bus interface module into the primary backplane via the EQL Editor.

In the context of equipment list editing, all nodes on the **remote bus** leaving the **BKF 102** are entered in a **”virtual”** subrack (VSR). This includes branches to the installation remote bus (via bus terminals) or to A120 components (via DEA202). The extent of the installation remote bus node must be taken into account when designating the ”slots” for the remote bus in the VSR. A linked group of A120 modules over the DEA 202 module is treated as **one** node on the remote bus.

The total number of nodes for a station, incl. remote bus, is limited (including A120 components).

☞ **Note** Exceeding the limit is currently reported in an appropriate error message when Exchange Online of the largest PB takes place. The basic requirement for starting AKF125 is a main memory of at least 590 KByte. There is an appropriate message to that effect.

The possible number of components in the A120 subrack is limited to 31 MWs. Typically, 2 MWs are occupied per module. If this number is exceeded a message is output. For details refer toTabelle 36, page 345.

- Possible nodes on the **remote bus** are (see also page 341):

  - Bxx yy6 zz modules (TIOs)
  - DAP 644, DAP 654, DAP 664
  - DEP 608, DAP 608 (different connectivity from DAP 6x4))
  - Bus terminal BKI 601 as remote bus branch to the **installation remote bus**
  - Interface module DEA 202 for coupling A120 subracks DTA 20X

- possible nodes on the **installation remote bus** are (see also page 345):

  - DEP 608, DAP 608
  - DAP 644, DAP 654, DAP 664 (different connectivity from DXP 608)
  - Bxx yy6 zz modules (TIOs) (different connectivity from DXP 608)

Scroll through the various subracks with the aid of <PgUp> / <PgDn>. You can call stored, pre–configured subracks or VSRs with <Ctrl.+PgUp> and <Ctrl.+PgDn>.

Facilities for capturing and visualizing status and diagnostic messages (BKF 102) include the SFB BKF1 in modified form (as compared to the application with BKF 101) and the PLC Diagnostics \ I/O System function in the Online main menu.

3.3.4.1 General information

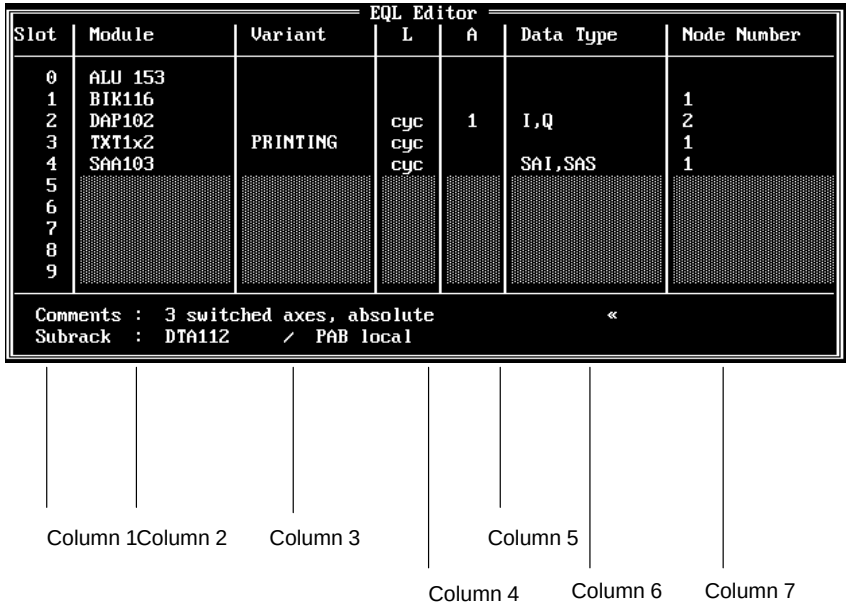
☞ **Note** For information about slot referencing, expansion limits (maximum number of modules) etc., please refer to the chapter ”Configuring the Hardware” in the A250 or A120 user manual.

The equipment list must be edited before the block because a plausibility check will be carried out. Input and output signals and near data structures can only be entered in the user program once you have entered the appropriate module in the equipment list.

The user can enter or amend the following parameters in the equipment list:

- Subrack types with nodes (I/O modules, Experts, ...)
- ALU type and its presettings
- I/O access procedure or group numbers (InterBus only), switch–off mode and timeout
- Logical Slot Numbers (Node Numbers)
- Phase and Cycle of Node and Job Elements
- Variants (software identifier) of standard and intelligent function blocks (Experts).
- User Comment
- Segmentation of the User Marker Area

After the equipment list editor is chosen, the following window will appear



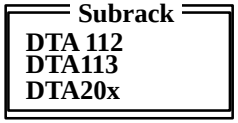
The equipment list editor consists of the following columns:

- Slot reference (SP)
- Node (Module)
- Variant
- Column 4, optionally A250 / InterBus S
  - Schedule Type (Z)
  - Group number (G)
- Switch–off Mode (A)
- Data Type
- Node Number (TN No.)

When the EQL editor is called up and the equipment list has been initialized, subracks are offered for entry.

☞ **Note** When configuring for InterBus nodes, you are assisted by being offered an appropriate subrack, also known as a ”virtual” subrack. The number of ”slots” can therefore be freely chosen and column 4 takes on another significance (group number). Nodes that are functionally similar can be grouped together by designating a common group number, which for instance allows them to be deactivated **together**. (see also page 325)

The following pulldown menu appears:



☞ **Note** When using ALU 151 .... 154 (A250), choose subrack DTA112 or DTA 113; when using ALU 204 / ALU 205 (A120), choose subrack DTA 20x.

In general you can bypass the pulldown menu by pressing <Ctrl>+<reference character>. The following special keys can also be used in the equipment list editor:

<Esc>	Break (Without Saving)
<→>, <Tab>	Next parameterizable column
<←>, <Shift-Tab>	Previous parameterizable column
<Home>	Firts column of the current slot reference (SP)
<End>	Last column of the current slot reference (SP)
<Ctrl>+<Home>	First slot reference in the current subrack.
<Ctrl>+<End>	Last slot reference in the current subrack
<PgUp>	Next subrack
<PgDn>	Previous subrack
<Ctrl>+<PgUp>	First subrack
<Ctrl>+<PgDn>	Last subrack

3.3.4.2 Possible modules and their parameters

Column 1 (SP)
Column 1 contains the number of the slot address, being either

- a real A250 subrack
  or a
- freely definable virtual subrack (InterBus).

Column 2 (Module)
This column is where you input individual nodes (modules).

The ALU is always located at slot reference 0. Press <Return> to open the following pulldown menu for configuring the ALU:

Enter

- ”Edit”, ”Equipment list”, Column 1 (ALU), <Return>, ”Enter”

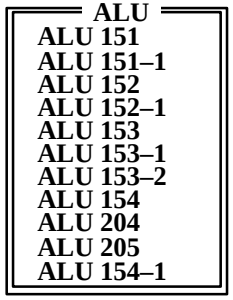
This menu item allows you to make entries about the ALU type and floating point processing in the equipment list editor.

If you do not want the data to be accepted, cancel your changes by pressing <Esc>.

ALU type

- ”Edit”, ”Equipment list”, Column 1 (ALU), <Return>, ”ALU type”

This is where you enter the ALU which is located in slot 0. Select the ALU you require from a list which you call by pressing <Return>.



Floating point

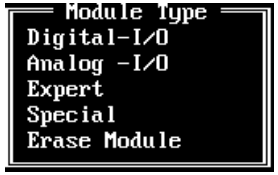
- ”Edit”, ”Equipment list”, Column 1 (ALU), <Return>, ”Floating point”

This is where you can specify whether you wish to carry out floating point processing with or without the arithmetic processor. This choice is only possible for those ALUs equipped with an arithmetic processor. In this case you can toggle between ”Emulation” and ”Hardware”. The default ”Emulation” can otherwise not be changed.

Table 2 Floating point

ALU type	Arithmetic proc.	Floating point
ALU 151	no	Emulation
ALU 151-1	no	Emulation
ALU 152	no	Emulation
ALU 152-1	yes	Emulation/Hardware
ALU 153	no	Emulation
ALU 153-1	yes	Emulation/Hardware
ALU 153-2	yes	Emulation/Hardware
ALU 154	yes	Emulation/Hardware
ALU 204/205	yes	Emulation/Hardware
ALU 154-1	yes	Emulation/Hardware

The I/O modules and intelligent function blocks are located at slot references 1 ... n.Press <Return> to open the following pulldown menu for configuring modules:



Via this menu you will reach the modules which can be entered in the equipment list. Only the modules which can be entered in the current slot address (SP) are shown. The modules are functionally arranged. You can delete nodes via this menu or with <Del>.

Both automation systems (A120 and A250) have different modules at their disposal.

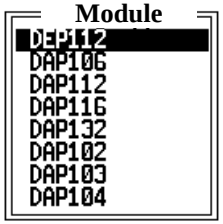
☞ **Note** Press <Ctrl>+<Return> to open the menu with the EQL editor functions (description after the last column).

Digital I/O

- ”Edit”, ”Equipment List”, Column 2, <Return>, ”Digital I/O”

You can enter the following digital I/O modules:

- A250 with DTA 112 / DTA 113



DEP112 Digital Input 32 Bit isolated
DAP106 Digital Output 16 Bit
DAP112 Digital Output 32 Bit isolated
DAP116 Digital Output 16 Bit
DAP132 Digital Output 32 Bit
DAP102 Digital Output 16 Bit / Digital Input 16 Bit
DAP103 Digital Output 16 Bit / Digital Input 16 Bit
DAP104 Digital Output 8 Bit / Digital Input 8 Bit

- A120 modules
  - A250 expansion using A120 modules with DEA201 in the DTA 20x
    or
  - A120 with ALU 204 / 205 in the DTA 20x

DAP212	Digital Output 4 Bit / Digital Input 8 Bit
DAP220	Digital Output 8 Bit / Digital Input 8 Bit
DEP216	Digital Input 16 Bit
DEO216	Digital Input 16 Bit
DEP220	Digital Input 16 Bit
DEP214	Digital Input 16 Bit 60VDC
DEP215	Digital Input 16 Bit 5VDC
DEP217	Digital Input 16 Bit 24VDC negated
DEP218	Digital Input 16 Bit
DEX216	Digital Input 16 Bit
DEP208	Digital Input 8 Bit
DEP210	Digital Input 8 Bit
DEP211	Digital Input 8 Bit
DEP209	Digital Input 8 Bit
DAP204	Digital Output 4 Bit
DAP208	Digital Output 8 Bit
DAP209	Digital Output 8 Bit
DAP210	Digital Output 8 Bit
DAP216	Digital Output 16 Bit
DAO216	Digital Output 16 Bit
DAP218	Digital Output 16 Bit
DAX216	Configureable Digital Output 16 Bit
DAP217	Digital Output 16 Bit current sink

Digital inputs and outputs are also located on DEA-H1 and DEA-K1. These are entered by using the function ”Enter Subrack”, since they are operated over Modnet 1/SFB.

❒ InterBus digital I/O modules (linked group over BKF 102)
For details of allowed modules see page341 on the subject of appropriate cabling.

AnalogI/O

- ”Edit”, ”Equipment List”, Column 2, <Return>, ”AnalogI/O”

You can enter the following analog I/O modules:

❒ A250 with DTA 112 / DTA 113

Module		
ADU 115	ADU115	32 analog inputs, isolated
ADU 116	ADU116	16 analog inputs, isolated
DAU 104	DAU104	Analog input / Analog output
DAU 108	DAU108	8 analog outputs, isolated
DAU 109	DAU109	8 analog outputs, isolated
DXX 116	DXX116	”Joker” module for entering third–party modules.

❒ A120 modules
- ❑ A250 expansion using A120 modules with DEA201 / DEA202 in the DTA 20x
  or
- ❑ A120 with ALU 204 / 205 in the DTA 20x

Module	
ADU 204	4 analog inputs (PT100 / +-0,5V)
ADU 205	4 analog inputs (+ - 10V / 20mA)
ADU 206	4 analog outputs (+ -1V/5V/10V)
ADU 210	4 analog inputs (+ -5V/10V/20mA)
ADU 214	Analog input 15 bits + sign
ADU 216	Analog input 16 bits
DAU 202	2 analog outputs, isolated
DAU 208	8 analog outputs, isolated
DXX 216	Freely configurable I/Os

❒ InterBus analog modules (linked group via BKF 102)
For details of allowed modules see page 341.

Experts

- ”Edit”, ”Equipment List”, Column 2, <Return>, ”Experts”

You can enter the following intelligent function blocks in the primary subrack and secondary backplane:

❒ A250 with DTA 112 / DTA113

Module	
ZAE 105	
SAI 103	
SAA 103	
VIP 101	**1)**
KPO 104	
POS 102	**2)**
POS 112	**3)**
POS 104	**4)**
POS 114	**5)**
DEZ 161	
TXT 1x2	
KOS 140	
KOS 141	

NOK116

ZAE105	Intellig. Module with 5 counters
SAI103	Intellig. function block switched axis pos. incremental
SAA103	Intellig. function block switched axis pos. absolute
VIP101	Intellig. function block Operation and Display Technique
KPO104	Intellig. function block Object Identification
POS102	Intellig. func. block Positioning: 2 axes incremental
POS112	Intellig. func. block Positioning: 2 axes absolute (SSI)
POS104	Intellig. func. block Positioning: 4 axes incremental
POS114	Intellig. func. block Positioning: 4 axes absolute (SSI)
DEZ161	Intellig. function block Interrupt (Z-subrack only)
TXT1x2	Intellig. function block Word Processing
KOS140	Intellig. function block Communication
KOS141	Intellig. function block Communication
NOK116	Intellig. function block Cam control; can be enabled by an appropriate entry in file EAKARTEN.STD.

1) VIP 101 also stands for components VIP101-1 and VIP101-2
2) POS 102 – via variant entry -01 / -02 – stands for POS102-01 and POS102-02
3) POS 112 – via variant entry -01 / -02 – stands for POS112-01 and POS112-02
4) POS 104 also stands for components POS 104-01
5) POS 114 also stands for components POS 114-01

❒ A120 with ALU 204 / 205

ZAE105 Intellig. function block with 1 counter
ZAE204 Intellig. function block with 4 counters
TXT2E1 TXT201 in Print Mode
TXT2c1 TXT201 in Networking Mode

The POS modules cannot be entered directly into the equipment list. Two ZAE 201s have to be entered for every POS.

☞ **Note** Only certain node numbers are permissible for intelligent function modules:
ZAE 201 1 ... 656
ZAE 204 1 ... 656
You then enter the number in the user program under parameter ”TN” for the FBs or specific SFBs.

Special

- ”Edit”, ”Equipment List”, Column 2, <Return>, ”Special”

You can enter the following special modules in the primary subrack:

❒ A250 with DTA 112 / DTA113

Module	
BIK 114	Modnet 1/SFB interface without power supply
BIK 116	Modnet 1/SFB interface with power supply
DNP 106	Power supply
DNP 116	Power supply
KPH 141 **)	Ethernet interface (Sinec H1)
KPP 131	Profibus Controller
KPP 132	Profibus Controller
KPE 141 **)	Ethernet interface MMSE
KPM 111	Controller for MAP3
BKF 102 *)	Interface module for InterBus (Modnet 1/IS)

*) Unlike the BKF101, an SFB of the BKF1 type is not necessary during normal operation (without Status, Diagnostics).

**) The appropriate ”–1” version can also be used as a HW component.

You can enter the following special modules in the secondary backplane:

Module	
DEA 106	Interface card for I/O bus coupling
DEA 116	Interface card for I/O bus coupling with augmented power pack

❒ A120 with ALU20x

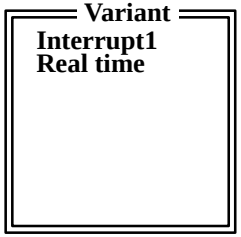
Module	
KOS 201	Modnet 1/SFB interface
KOS 202	Modnet 1/SFB interface
KOS 203	Modnet 1/SFB interface
KOS 204	Modnet 1/SFB interface
KPP 231	Profibus Controller
DNP 210	Power supply
DNP 205	Power supply

☞ **Note** ALU 204/205 requires that node 1 in the subrack be a power supply (DNP 205 or DNP 210).

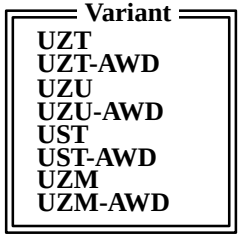
☞ **Note** The KOS 201 can only be assigned to one of the first three nodes.

Column 3 (Variants)
Some modules (e.g. TXT 1x2, DEZ161, KOS140, KOS141) are capable of performing different functions with the aid of various loadable basic software programs. These functions can be selected in the column ”Variants” via a menu.

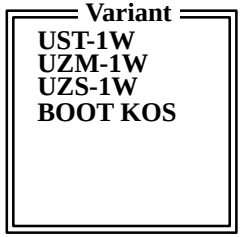
Example for DEZ161



Example for KOS140



Example for KOS141



If a module is entered that differentiates between variants, one of these will be prompted as a suggestion, and can be changed if necessary.

If a module does not differentiate between variants, the ”Variants” column will remain empty and cannot be edited.

Each variant has a particular feature that is driven by appropriate SW. This is also known as a software identifier.

Column 4 (Z), (G for InterBus))

❒ Z

This column indicates whether the modules of this line are operated with cyclic scheduling or with direct scheduling. In the case of cyclical scheduling, modules can still be directly scanned.
InterBus recognises cyclic processing only; the purpose of this column is therefore to build groups of individual modules or DTA20x subracks.

Press <Return> to toggle between ”cyc” (cyclic) and ”dir” (direct) in this column.

Cyclic

Cyclic scheduling means that the module is scanned in each cycle or in each n–th cycle.

It is generally possible to reduce the the scan time by means of cyclical scheduling. This is achieved by not scanning each module in every cycle.

Cyclic scheduling requires you to parameterize the job elements (for further information about cyclic processing and job elements, see ”Equipment List Editor / Job elements” below).

Direct

Direct scheduling means that the node number (TN No.) of a module is not automatically called at the end of a scan; instead it must be directly called in a network. The module is only polled at that point in time. If the TN No. is never called, the module is never scanned.

For this call there are differences between intelligent function modules and input/output modules.

The TN number is noted in the SFBs of the intelligent function blocks. You are therefore automatically asked to input the TN number during configuration.

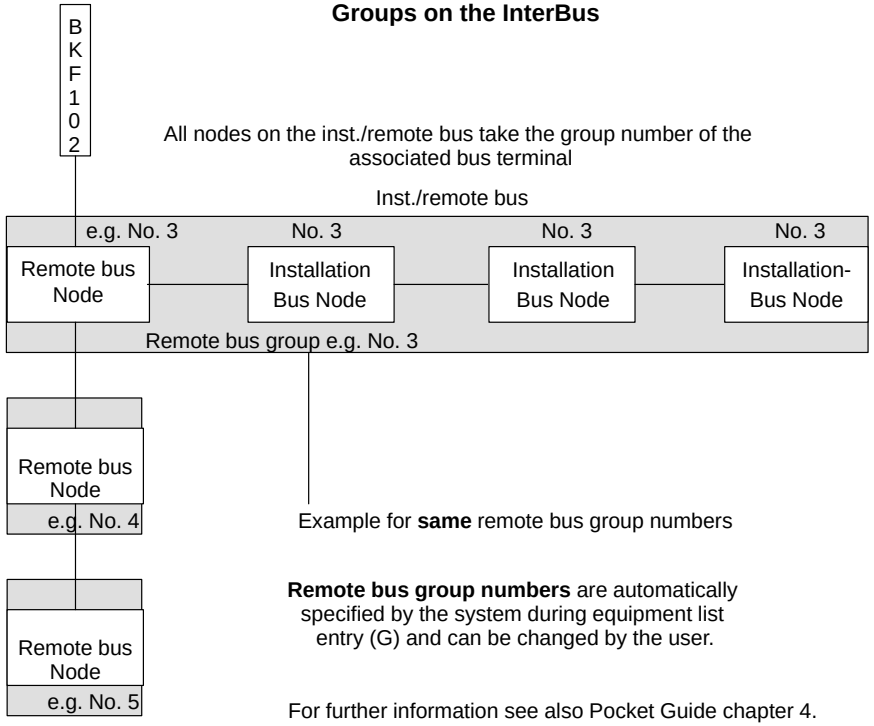
The TN number is not noted in the SFBs of the I/O modules. The TN numbers must be called up here with the SBFs for input/output (IN, OUT, ANIN and ANAOUT), before the inputs and outputs can be interlinked. This call allows the scanning of complete I/O modules, individual I/Os or I/O ranges.

☞ **Note** I/O points on the InterBus modules can be triggered **bit–by–bit**.

Caution There is no syntax check on whether the TN No of a directly scheduled I/O module will be called in a network. Inputs or outputs which are never processed may be interlinked accidentally.

❒ Column 4: G (for InterBus)
Automatic group number entry. Each node on the **remote bus** is first assigned its own group number. The nodes on the **installation remote bus** always take the group number of the associated bus terminal.
Group numbers can be **altered** by the user. (Several bus terminals can also be assigned a shared group number.)
For instance, if a module from a group defined by this means fails, all the modules in the group are shut down by the system. For further information see user manual InterBus Components for Modicon TSX A250.

Groups on the InterBus



Remote bus group numbers are automatically specified by the system during equipment list entry (G) and can be changed by the user.

For further information see also Pocket Guide chapter 4.

Column 5 (A)
In this column you can define the switch–off behavior of the modules.

This function parameterizes the behavior of the outputs from the PLC on program halt. It is unimportant whether the stop occurs as a result of an operating function or a program run–time error.

You can use <Return> to toggle between ”0” and ”1”.

- 0 means that the status of the signals remains unchanged

- 1 means the signals are terminated (i.e. reset to ”0”)

The behavior parameterized in this manner is only effective after the program has been loaded into the PLC. The current behavior mode in the PLC can be displayed via ”Online”, ”PLC status”.

Column 6 (Data type)
”Data Type” displays the data structures / operands which can be used with the current module.

Column 7 (TN No.)
Each module is assigned a **software** ”**slot number or node number**” (in part these are to be assigned to the actual operands of SFBs).

It is generally the case that these TN numbers are designated by module type. Each group has its own permitted number range:
Special (e.g. BIKxxx) module: 1 ... 9
I/O module: 1 ... 656
Each number in **both** these number ranges can be allocated **once** only.
The Expert modules shown below all have their **own** number range; this means that the same TN number(s) can be used for each Expert type.

ZAE 105	1 ... 63	
SAI 103	1 ... 63	
SAA 103	1 ... 63	
VIP 101	1 ... 10	
POS 102/112	1 ... 31	**)
POS 104/114	1 ... 31	**)
DEZ 161	1 ... 99	
TXT 1x2	1 ... 50	
KOS 140	1 ... 50	
KPO 104	1 ... 50	**)
KOS141	1 ... 50	
BKF102	1 ... 4	
NOK116	1 ... 5	

Enter these node numbers in the user program beim Parameter ”TN” der FBs oder spezifischen SFBs ein. The Experts highlighted with **) are nodes with a common or overlapping range.

The following exceptions apply when expanding with A120 modules:
Common number ranges and their I/O modules:

KPP 231	1 ... 656
ZAE 201	1 ... 656 *)
ZAE 204	1 ... 656 *)

Common number ranges:

TXT 2E1	1 ... 31 *)
TXT 2C1	1 ... 31 *)

Common number ranges:

KOS 201	1 ... 31 *)
KOS 202	1 ... 31 *)
KOS 203	1 ... 31 *)

KOS 204 1 ... 31 *)
Modules highlighted with * can only be operated in the master subrack with ALU204/205.

The I/O addresses of the SYM/COM block and user program refer to this **software** number and **not** to the **hardware** slot number in the first column (”Slot”). The software thus becomes **slot–independent**;”allocation” within the equipment list is possible.

Thus it is possible to use the same program on different hardware configurations. They adapt to the number of the module of the user program (in this column). It is not necessary to adapt the user program to the slot addresses (slot).

As a default, the slot number is given for I/O modules and a sequential number for intelligent function blocks.

Erase Module

- ”Edit”, ”Equipment List”, Column 2, <Return>, ”Erase Module”

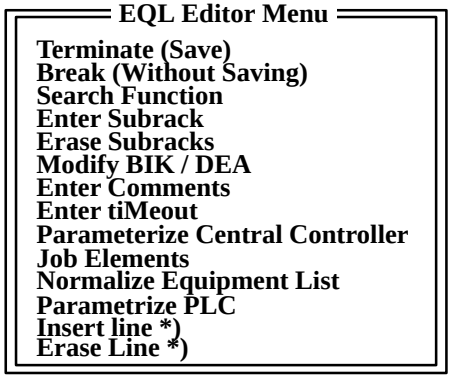
The node on which the cursor is positioned will be deleted.

You can only delete a BIK when all subracks which are connected to this network have been deleted. If a DEA is deleted, then all modules of this subrack will also be deleted.

Caution If modules are to be erased from the equipment list, all symbols and comments concerned must first be deleted from the SYM/COM file.

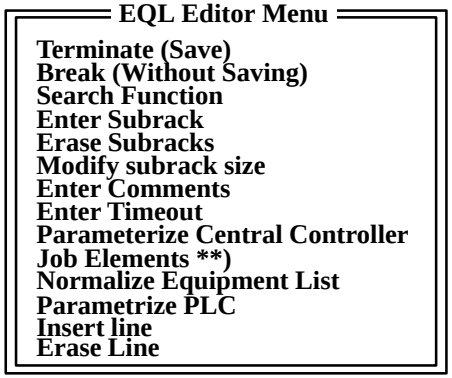
3.3.4.3 Equipment List editing functions

Press <Ctrl>+<Return> to call the functions of the equipment list editor. If the cursor is on an **A250 subrack** the following menu is offered:



***) for InterBus only**

If the cursor is on a **virtual subrack** (VSR) the following menu is offered:



****) for A250 subrack only**

Equipment List Editor / Terminate (save)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Terminate”

This function allows you to end the entry and save the changes. In addition, the following files will be created:

- BES25.IMG Loadable Equipment List for the Firmware
- SSPTAB25.DAT Allocation and Addresses of Markers and System Markers in Signal Memory

The equipment list editor can only be terminated if it has been possible to bring parameterization of the controller correctly to an end. If this is not the case, an appropriate message is shown.

Equipment List Editor / Break (Without Saving)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Break (Without Saving)”

This function is used to abort the entry and exit the editor.

Break is equivalent to using the <Esc> key outside the pulldown menu.

Equipment List Editor / Search Function

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Search Function”

This function allows you to search for slot addresses and modules.

- Search for Slot Address: If you enter a number under ”Search for”, a slot search will be activated. The number entered will be taken to be the slot to be located and displayed. If the required slot does not exist, the next possible slot will be shown.

❒ Search Module: Enter the name of the desired module under ”Search for”; it is then searched for starting from slot address 0. With a leading ”+” the search is carried out forward; with a leading ”-” the search is carried out backward. e.g. +DAP112

Equipment List Editor / Enter Subrack (Modnet 1/SFB)

This function is used to enter the next subrack. Depending on the current cursor position, **different** windows are offered for subrack expansion. A distinction is made between

- subracks for expanding A250 with A250 or A120 via **Modnet 1/SFB.**

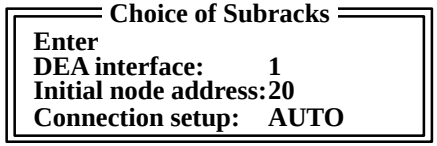
and

- ”virtual” subracks (VSRs) for expanding A250 with A120, TIO (IP20) and DEP 6xx or DAP 6xx modules (IP65) via **InterBus (Modnet 1/IS)**.

Subrack expansion via Modnet 1/SFB

”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
Enter Subrack”

- If the cursor is on a module with an **MN 1/SFB** interface, the following window opens:



Equipment List Editor / Start Entry (MN 1/SFB)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
Enter Subrack”, ”Enter”

Use this function to enter new subracks. Proceed as shown in the following **steps**:

Equipment List Editor / DEA interface (MN 1/SFB)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”DEA interface”

Step 1 Input ”DEA interface”:
The DEA interface address (slave) to which the secondary backplane is to be connected must be entered here (1....15). The input must agree with the HW setting on the ”DEA”. A maximum of 8 secondary backplanes and 7 DEA-H1/K1 are allowed.

Equipment List Editor / Initial address (MN 1/SFB)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Initial node address”

Step 2 Input ”Initial node address”:
Enter at this point the starting node address from which the 10 slots on this subrack are to begin. Addresses can be freely designated in increments of 20 within the range 20 to 980.

Note Gaps can be freely configured.

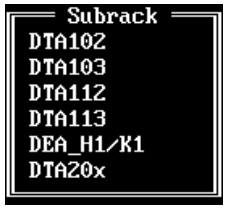
Equipment List Editor / Choice of networking structure (MN 1/SFB)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Initial node address”

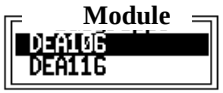
Step 3 Input ”Connection setup”:
This is where the user can define whether the bitbus interface is to be set up automatically (AUTO) or via the SFB DEAKOP (HAND), that is, **without** an error message, e.g. by program
event, or manually.

Step 4 Input ”Enter”.
Both these particulars must be accepted via menu item ”Enter”. You can also break off by pressing ”Escape”.

Reaction A new menu will be displayed so that you can input the subrack type:
for A250 (DTA112, DTA113, ...)
for A120 (DTA20x).



The bus networking can then be chosen from another module menu (in the case of A250 subracks).



☞ **Note** If you choose the A120 magazine DTA20x, a subrack with 16 free slots and a previously entered networking module DEA 201 is displayed for further processing.

☞ **Note** Subracks are always entered in increments of 20. The BIK you use in the first entry is occupied with the next possible slot address. If the BIK on slot address 1 is selected before the ALU BIK, the following happens:
The BIK on slot reference 1 supplies slot references from 20 onward; the BIK on the ALU with slot reference 0 supplies slot references from 40 onward.

Having made your entry, choose the subrack type you require. You can choose between:

If DEA_H1/K1 is selected as a subrack, the Modnet 1/SFB interface will not be chosen. In addition, the equipping of this subrack takes place automatically and you cannot change the modules entered. DEA_H1/K1 consists of DEA-H1 or DEA-K1.

Subrack expansion for InterBus (Modnet 1/IS)

Equipment List Editor / Enter ”virtual” subrack

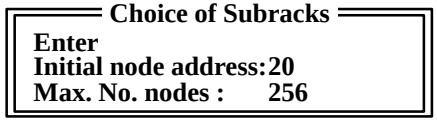
”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
Enter Subrack”

With the cursor on BKF102, use this function to enter **one** ”virtual” subrack (VSR) per BKF102. A maximum of 4 x BKF102 modules can be entered.

☞ **Note** The installation remote bus can be connected with InterBus modules over a BKI 601 bus terminal within this VSR.

☞ **Note** By the ”virtual” entry of a **DEA 202** interface module (the module is actually located in the DTA20x A120 subrack) the remote bus is expanded to permit connection of a DTA 20x subrack.

❒ If therefore the cursor is on the **BKF102** module (InterBus adapter) the following window opens:



Equipment List Editor / Start entry (Modnet 1/IS, InterBus)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”

Use this function to enter a ”**virtual**” subrack.

Note The VSR can contain 256 nodes on the remote bus.

Note The available slot address range is 20 to 999. The highest definable initial address is 980.

To enter a VSR proceed as follows:

Equipment List Editor / Initial InterBus node address

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Initial node address”

Step 1 Input ”Initial node address”:
Enter here the starting node address from which a maximum of 256 ”slots” for this VSR can be defined. Addresses can be freely defined in increments of 20 within the range 20 to 980 in the whole system.

Note Slot gaps can be freely configured.

Equipment List Editor / Max. No. nodes (Modnet 1/IS, InterBus)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Max. No. nodes”

Step 2 Input ”Max. No. nodes”:
At this point enter the number of ”slots” required on the remote bus, including branches via the bus terminals, for connecting the IP20, IP65 and A120 components.

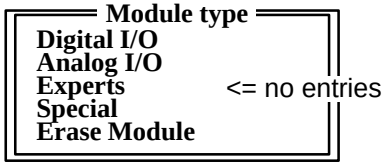
Step 3 Input ”Enter”.
Both these particulars must be accepted via menu item ”Enter”. In addition, you can break off by pressing ”Escape”.

Reaction The VSR complete with its specified ”slot” data is offered for further processing.

Equipment List Editor / InterBus module type selection

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”, **<Return>**

Step 4 When you press <Return>, a global window for selecting the modules to enter is displayed:



Equipment List Editor / InterBus, Digital module selection

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”, <Return>, **<Return>**

Binary modules for the remote bus VSR

Module
BDI 346 00
BDO 346 00
BDI 356 00
BDO 356 00
BDM 346 00
BDM 346 30
BDI 546 50
BDI 746 50
BDO 946 50
DEP 608
DAP 608
DAP 644
DAP 654
DAP 664

BDI346 00	Input 2 x 8, pot.–isolated, 24 VDC, initiator mode
BDO346 00	Output 2x8, pot.–isolated, 24VDC, actuator mode
BDI356 00	Input 2x16, pot.–isolated, 24 VDC, initiator mode
BDO356 00	Output 2x16, pot.–isolated, 24 VDC, actuator mode
BDM346 00	16 x E, 2x16A, 24 VDC, initiator/actuator mode
BDM346 30	8 x E pot.-linked, 2x4 A pot.–isolated, 24 VDC
BDI546 50	Input 2 x 8, pot.–isolated, 120 VAC, initiator mode
BDI746 50	Input 2 x 8, pot.–isolated, 230 VAC, initiator mode
BDO946 50	Output 1 x 16, pot.–isolated, 24 ...230 VAC, actuator
DEP 608	Input 1 x 8, pot.–isolated, 24 VDC, inst.-remote bus***)**
DAP 508	Output 1x8, pot.–isolated, 24 VDC, inst.-remote bus***)**
DAP 644	2 x 8E, pot.–isolated, 2 x 4A, 24 VDC, without flange
DAP 654	2 x 8E, pot.–isolated, 2 x 4A, 24 VDC, PG9 cross–grad.
DAP 664	2 x 8E, pot.–isolated, 2 x 4A, 24 VDC, initiator socket

***) Modules may naturally be entered in the VSR for the remote bus, but on the cabling side they only go with the ”installation remote bus”. This circumstance is highlighted by an appropriate message when you make an entry in the VSR.**

Equipment List Editor / InterBus, Analog modules selection

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”, <Return>,Analog I/O selection, **<Return>**

Analog modules for the remote bus VSR

Module
BAM 096 00

BAM096 00	4 x E, 2 x A, PT100 mode

Equipment List Editor / InterBus, special modules selection

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”, <Return>,Special selection, **<Return>**

Special modules for the remote bus VSR

Module	
BKI 601	BKI 601 bus terminal as branch to installation remote bus
DEA 202	DEA 202 interface card for connecting a DTA 202 to the remote bus

The bus terminal entry is highlighted with a filled cursor field (■) in the ”Module column”.

Equipment List Editor / Erase Module (MN 1/IS, InterBus)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”, <Return>**,Erase Module**, <Return>

During deletion of **bus terminals**, the inst.-remote bus modules are also removed after several requests for confirmation.

Equipment List Editor / InterBus, determining ”VSR” length

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”, <Return>,Special selection, <Return>,<Ctrl.+Return>**,”Enter Subrack”**, <Return>

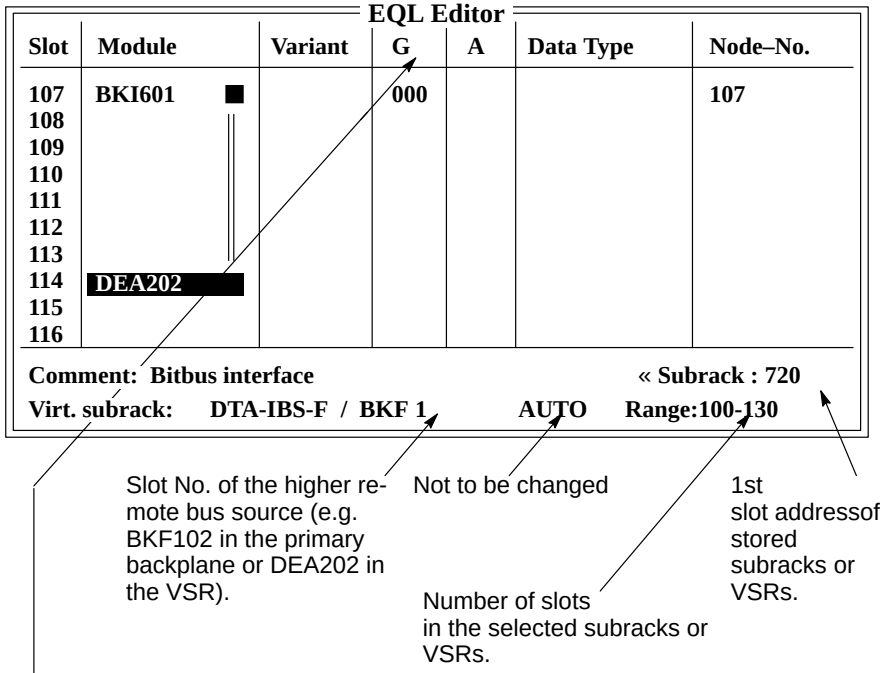
Enter	
Last Slot:	**113**

Note The maximum number of ”slots” required on the installation remote bus **must** be entered right after the bus terminal information.

The **highest** possible slot number is calculated by: current BKI slot address + 41.

When the highest slot has been input via ”Enter”, the defined slot number range is highlighted in the ”module column” with a double line.

Equipment list elements with InterBus



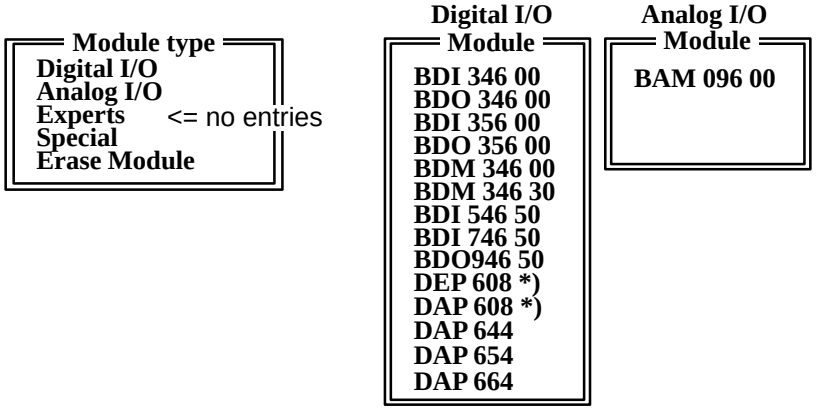
For InterBus mode, the meaning of the **4th column** changes (since a virtual subrack exists). The group number is entered / modified in column G. (See also page 325)

Slot:	Virtual slot as config. aid for the module position on InterBus
Virt. subrack:	Unlike a real subrack, a virtual subrack (VSR) is used as a configuration aid. (The number of slots is definable.)

Equipment List Editor /"Slot" allocation for the inst. remote bus

- "Edit", "Equipment List", <Ctrl>+<Return>, "
  Enter Subrack", "Enter", <Return>,Special selection, <Return>,
  <Ctrl.+Return>**,"**Enter Subrack**"**, <Return>, <Return>, Module type
  selection, **Module selection (Digital Analog)**

For entering **inst. remote bus** modules, the modules shown below are displayed via the module type selection:



Note Only modules highlighted with *) may be entered **without** restriction in the equipment list (**inst. remote bus**). This circumstance is highlighted by an appropriate message when you enter inappropriate modules in the inst. remote bus.

Slot	Module	Variant	G	A	Data Type	Node–No.
107	BKI601		000			107
108	DEP 608					
109	DAP 608					
110	DEP 608					
111	DAP 608					
112	DEP 608					
113	DAP 608					
114	BKI 601					
115	DEP 608					
116	DAP 608					

Comment: Bus terminal, installation bus «
Virt. subrack: DTA-IBS-F / BKF 1 AUTO Range:100-130

Equipment List Editor / ”A120 on the inst. remote bus”

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”
  Enter Subrack”, ”Enter”, <Return>,Special selection, <Return>,<Ctrl.+Return>**,”**Enter Subrack**”**, <Return>, <Return>, Module type selection, **Module selection (DEA 202)**

On selecting the ”special” module, interface module DEA 202 is entered as a dummy or ”connection reminder” for a DTA 202 subrack. The DTA 202 is then positioned by keyboard entries

<Ctrl.+Return>, ”Enter Subrack”,<Return>

followed by designating the initial address of the modular A120 subrack.

Enter
Initial node address: 720

The result of the entry appears as in the example: The DEA 202 interface module is already entered in the subrack.

Slot	Module	Variant	G	A	Data Type	Node–No.
720	DEA 202		000			720
721						
722						
723						
724						
725						
726						
727						
728						
729						

Comment: Bitbus interface «

Subrack: DTA 20x / DEA 202 112 AUTO Range: 720-738

A subrack with 18 free slots for entering digital and analog modules is offered. The customary A120 modules can be entered.

Note Subracks are input in incremental steps of twenty.

Caution The maximum number of modules that can be entered must be adjusted to match the call–handling capacity of the InterBus. This means that modules can continue to be added until the maximum extent of I/Os in an A120 expansion is reached (with DEA 202 coupling), described by a maximum number of 31 MWords for each I/O.

Note When you enter modules in the equipment list, the above limits are monitored.

The I/O memory utilization of the individual A120 I/O modules is as follows:

Table 3 I/O memory utilization per Module on the InterBus

Module name	I–Marker Words	Q–Marker Words
DEP 214, 215, 216, 217, 218, 220, (296, 297)	1	
DEO 216, DEX 216	1	
DEP 208, 209, 210, 211	1	
DAP 216, 217, 218, DAO 216, DAX 216		1
DAP 204, 208, 209, 210		1
DAP 212, 220, (252, 253, 292)	1	1
DXX 216	2	2
DAU 202		2
DAU 208		8
ADU 204, 205	5	
ADU 206, 216	5	1
ADU 210	5	2
ADU 214	9	4

Equipment List Editor / Erase Subrack

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ” Erase Subrack”

Use this function to delete the subrack currently indicated. Please remember that all modules entered will also be deleted. Confirmation is therefore required.

Note The inst. remote bus VSR is removed via menu item ”Erase Module”.

The primary backplane cannot be deleted; use the function ”Normalize Equipment List” for this purpose. If you wish to delete from the primary backplane a BIK to which secondary backplanes are connected, you must first delete all subracks connected to the BIK concerned. The same is true of a DEA 202 that has been entered on the installation remote bus.

Deleting subracks can lead to gaps in the continuous assignment of slot numbers. The gaps are filled by the entry of new subracks.

Equipment List Editor / Modify BIK / DEA (A250/A120 subracks only)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Modify BIK / DEA”

This function is used to subsequently modify the bit address connection or the DEA networking.

- Input BIK: Slot reference of the BIK in the primary backplane to which it is intended that the new subrack should be connected.

- Input DEA: The subrack number in accordance with the address setting on the DEA module. A maximum of 8 subracks and 7 DEAH1/K1 can be connected per BIK. The address setting must be in the range 1 to 15.

Equipment List Editor / Modify subrack size (VSR on the remote bus)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Modify subrack size”

This function is used to subsequently modify the size of the virtual subrack for the **remote bus**. The information is requested after the required number of nodes. It is not possible to change the size of the **installation remote bus**.

Note The installation remote bus cannot be modified with this function. To make such a change, delete the bus terminal and re–enter with higher values, if necessary using another slot on the remote bus.

Modification limits:
Virtual subrack size cannot be reduced below the highest assigned **remote bus** slot address.

Virtual subrack size cannot be expanded beyond the next assigned slot address.

Slot	Module	
100	**DEP 608**	**<——Initial address of remote bus (e.g.)**
101		
102	**DEP 60ß**	
103		
104		
105	**DAP 608**	
106		
107	**BKI601**	<——Initial address of installation remote bus
108	**DEP 608**	
109	**DAP 608**	
110	**DEP 60ß**	
111	**DAP 608**	
112	**DEP 608**	
113	**DAP 608**	
114		
115	**DAP 608**	
116		<——End address of installation remote bus
117		**<——Continuation of remote bus**
118	**DEA 202**	<——Dummy for A120 expansion on remote bus
119		
120	**DEP 608**	
:		
:		
130		**<——End address of remote bus (e.g.)**
:		
:		
139		
140	**DEA 202**	**<——Initial address for next subrack (e.g.)** **(also next possible init. address)**

In the above example, a reduction down to slot 120 is possible. Expansion as far as slot 139 is available.

Equipment List Editor / Enter Comments

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Enter Comments”

This function allows you to enter comments for a slot reference in the lowest line of the editor. Normally a preset comment appears which describes in abbreviated form the function of the module entered. The comment is valid for the line on which the cursor is positioned. Up to 40 characters of comment are possible.

Equipment List Editor / Enter Timeout

Note No entry can be made in the case of InterBus nodes.

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Enter Timeout”

This input determines the timeout time (in milliseconds) after which all the DEA output modules in a subrack are switched off.

Values from 0 to 2 550 ms can be entered. The values are rounded off to the nearest 10 ms.

Remember that timeouts cannot be parameterized for the primary backplane.

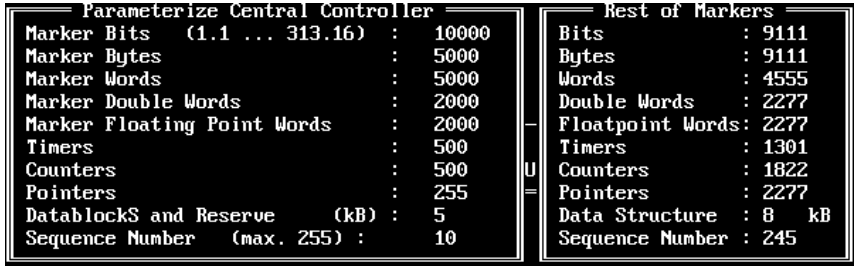
Equipment List Editor / Parameterize Central Controller

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Parameterize Central Controller”

The central processing unit is the controller of the PLC. It executes the various program instructions in line with rules laid down by the basic software. In other words, it

- organizes the reading of external data and signals into the signal memory.
- processes these data and executes calculations,
- continually stores process results in the signal memory,
- organizes the output of results

Parameterization of the controller determines the segmentation of the signal memory in respect of the following signals:


Parameterize Central Controller
Marker Bits (1.1 ... 313.16) : 10000
Marker Bytes : 5000
Marker Words : 5000
Marker Double Words : 2000
Marker Floating Point Words : 2000
Timers : 500
Counters : 500
Pointers : 255
DatablockS and Reserve (kB) : 5
Sequence Number (max. 255) : 10
Rest of Markers
Bits : 9111
Bytes : 9111
Words : 4555
Double Words : 2277
Floatpoint Words: 2277
Timers : 1301
Counters : 1822
Pointers : 2277
Data Structure : 8 kB
Sequence Number : 245

A further window will show you the possible maximum values for each of the marker types. If the available space in the signal memory area is exceeded, negative values will be outputed during the remaining allocatin. In this case, the parameterization of the controller can not be terminated.

☞ **Note** The maximum values are dependent on the mounted component selection. Complex systems have less memory available for markers than those of minimal configuration.

The parameters you assign to the controller will be saved if you terminate with <Ctrl>+<B>. To break off without saving, press <Esc>.

☞ **Note** Regardless of the central processing unit (ALU), 64 kbyte markers, I/Os, etc. can always be parameterized. If the ALU is not in a position to process the desired number of signals, an error message is produced when the program is being loaded in the PLC. The number of signals used must then be reduced, or a bootload must be performed. The equipment list editor then allows entries up to the current maximum possible size.

System markers

In addition to the user defined markers, there are a number of system markers. They give the user information about the status of the PLC, timer information, generate flashes and offer additional useful information which can be evaluated by the user. Remember that system markers are only readable for users.

- SM System Marker Bit
- SMB System Marker Byte
- SMW System Marker Word
- SMD System Marker Double Word
- SMG System Marker Floating Point Word

Marker Bits

The following values are valid for marker bits

- Signal Type M
- Allocation in Signal Memory 1 byte
- Value Range 0 ... 1
- Number (maximum) 30 000
- Number (default) 10 000

Marker Bytes

The following values are valid for marker bytes

- Signal Type MB
- Allocation in Signal Memory 1 byte
- Value Range 0 ... 255 (-128 ... 127)
- Number (maximum) 30 000
- Number (default) 5 000

Marker Words

The following values are valid for marker words

- Signal Type MW
- Allocation in Signal Memory 2 bytes
- Value Range 0 ... 65 535 (-32 768... 32 767)
- Number (maximum) 30 000
- Number (default) 5 000

Marker Double Words

The following values are valid for marker double words

❒ Signal Type	MD
❒ Allocation in Signal Memory	4 bytes
❒ Value Range	0 ... 4 294 967 295 (-2 147 483 648 ... 2 147 483 647)
❒ Number (maximum)	15 000
❒ Number (default)	2 000

Marker Floating Point Words

The following values are valid for marker floating point words

❒ Signal Type	MG
❒ Allocation in Signal Memory	4 bytes
❒ Value Range	–3.4E+38 ... +3.4E+38
❒ Number (maximum)	15 000
❒ Number (default)	2 000

Marker Times (Timers)

The following values are valid for timers

❒ Signal Type	T (TAW, TSW)
❒ Allocation in Signal Memory	7 bytes
Division	T 1, TAW 2, TSW 2, internal file 2
❒ Number (maximum)	9 000
❒ Number (default)	500

Marker Counters (Counters)

The following values are valid for counters

❒ Signal Type	C (CIW, CSW)
❒ Allocation in Signal Memory	5 bytes
Division:	C 1, CIW 2, CSW 2
❒ Number (maximum)	12 000
❒ Number (default)	500

Pointers

The following values are valid for pointers

❒ Signal Type	P
❒ Allocation in Signal Memory	4 Bytes
❒ Value Range	0000:0000 (Hex) ... FFFF:FFFF (Hex)
❒ Number (maximum)	4095
❒ Number (default)	255

Data Blocks and Reserve

❒ Near Data Structure Blocks (Near–Data Structures, in the sense of near = quick):

These are data structure blocks defined in the data structure editor which have been provided with the addressing attribute ”near”. They will be stored in the signal memory if they are used in the user program. The advantage of quickly accessing them is coupled with the disadvantage that they are not exchangeable online. Near–data structures will be stored in signal memory as long as room is available.

❒ Reserve:

This data range is also used for the allocation of signal memory.

Caution The size of the signal memory can only be reduced and not increased. It is therefore important to always have a sufficient range of reserve available so that subsequent expansions will be possible.

If signal memory is to be made larger than its current size including the reserve, the system must be rebooted.

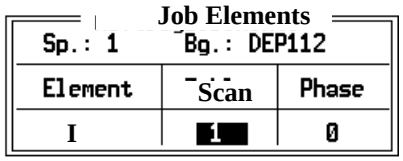
☞ **Note** No entry can be made in the case of InterBus nodes.

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Job Elements”

The requirement for this function is the entry ”zyk” in column ”Z”.

For modules exclusively with direct scheduling an announcement appears on selection of this menu item.

For each type entered in the ”Data type” column (job element) you can enter separate parameters. After selecting the function a menu with the following structure appears:



Inside this function you can use the following special keys:

<Ctrl>+<B>		Break (with Save); the values entered are accepted
<Esc>		Break (without Save); the values entered are ignored, i.e. the existing values remain unchanged
Arrow keys:	left, right, up, down	Choice the value to be edited
<Return>		Prepare line for input of values

Sp: / Bg:

Sp: Enter here the slot reference on which the cursor was positioned before this function was selected. Bg: designates the module located on this slot reference.

Element

”Job elements” means all those signals that are present in a module and that can be parameterized in AKF.

Specifically, this includes:

- Digital I/O: Input and output bits
- Analog I/O: Input and output words
- Experts: Actual and setpoint values, status

Scan and phase can be parameterized separately for each element type. A maximum of 50 scans and 50 phases can be parameterized.

Scan

The number entered at cycle determines the cycle interval in which this element is processed. That means if scan = ”5” the signal memory of the module concerned is polled every fifth user program cycle.

If you enter the value 0 for the cycle of a schedule definition, then this element is not cyclically, but directly scheduled. If all schedule definitions are directly scheduled, then the preset in column ”Z” is automatically set to ”dir” (direct).

Phase

The number entered at phase determines the number of dummy cycles at progam start before the cycle intervals are begun. Thus a phase delay is created in the cycle counter.

Example cycle / phase for a DAP106 on slot reference 4

Job Elements

Sp. 4	Bg. : DAP106	
Element	Scan	Phase
Q	5	0

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Scan	-	-	+	-	-	-	-	+	-	-	-	-	+	-	-

Job Elements

Sp. 4	Bg. : DAP106	
Element	Scan	Phase
Q	5	2

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Scan	-	-	+	-	-	-	-	+	-	-	-	-	+	-	-

+: yes -: no

Equipment List Editor / Standardize

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Standardize”

The equipment list is standardized, i.e. reset to a standard presetting.

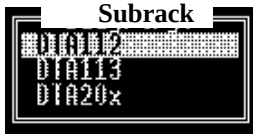
Note Only this function can be used to modify the primary backplane type.

This includes the following operations:

- Erase Subracks
- Erase all entered modules
- Set controller parameters to default values

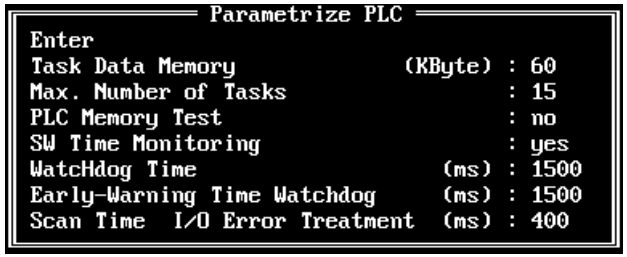
If you have standardized the equipment list by mistake, exit the equipment list editor by pressing <Esc> and call it again. Then your ”deleted” equipment list reapears.

After standardizing you can begin editing again. A menu is displayed for this purpose, and you can choose a primary backplane from it.



Equipment List Editor / Parameterize PLC

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Parametrize PLC”


Parametrize PLC
Enter
Task Data Memory (KByte) : 60
Max. Number of Tasks : 15
PLC Memory Test : no
SW Time Monitoring : yes
WatcHdog Time (ms) : 1500
Early-Warning Time Watchdog (ms) : 1500
Scan Time I/O Error Treatment (ms) : 400

Equipment List Editor / Enter

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Parametrize PLC”, ”Enter”

The modification will be accepted.

Equipment List Editor / Task Data Memory

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Parametrize PLC”, ”Task data memory”

The ”Task data memory” setting reserves part of the RAM in the address space of the first MB (a total of 768kB is available) for static data and task stack areas.

This setting, which is dependent on the user configuration, is not amendable online. For this reason the planned maximum configuration must be taken into consideration. Additional OBs can only be added via ”Exchange Online”, for example, if a sufficient reserve is available in ”Task Data Memory Range”. A setting that is too high reduces the memory available for data blocks.

The values 40 to 180 are permissible.

We recommend the following setting:

36k + (Number of BIKs)*3k + (Number of OBs without OB1)*8k.

If experts are used:	additional 9k;
if an SEAB interface is used:	additional 3k;
If MMSE is used:	additional 8k;

Equipment List Editor / Maximum Number of Tasks

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>,
  ”Parametrize PLC”, ”Max. Number of tasks”

Just as with ”Task data memory”, the ”Max. Number of tasks” setting has to be adapted to the user configuration:

15 + (Number of BIKs) + (Number of OBs without OB1).

If experts are used:	additional 3;
if a SEAB interface is used:	additional 1;
if MMSE is used	additional 2;

The values 15 to 50 are permissible.

Equipment List Editor / PLC Memory Test

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>,
  ”Parametrize PLC”, ”PLC memory test”

All loaded firmware program segments can be cyclically checked for version changes (read–test with comparison of checksum and CRC value). When this function is activated, scan times of OB1 and
time–controlled OBs running in timesharing-mode increase by about 20 %.

☞ **Note** We recommend **NO** for the standard setting. If a station is not running correctly, this function can check (when Memory test = **YES**) **whether** FW parts have been mixed (SM87=1) and **which** original FW was loaded in the PLC (SMW13).

Equipment List Editor / SW time Monitoring

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Parametrize PLC”, ”SW time Monitoring”

You can toggle between ”Yes” and ”No”.

Yes: SW time monitoring (watchdog) is active, meaning that the user program is halted when the SW watchdog time expires. This prevents the PAB from being switched off and the ALU from going into boot mode once the watchdog time has expired.

No: SW time monitoring is not active, meaning that once the watchdog time has expired the PAB is switched off and the ALU goes into boot mode.

Calculation of SW watchdog time:

SW watchdog time = specified watchdog time – 110 ms

(110 ms is the maximum inaccuracy.)

Note If ”SW watchdog: Yes”, then the smallest permissible watchdog time is 300 ms.

Equipment List Editor / Watchdog time

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>,
  ”Parametrize PLC”, ”WatcHdog time”

The scan times of the OBs will be monitored to see if they exceed the maximum values. Such excess scan times can be caused by, among other things, program loops or recursions with incorrect or missing abort conditions in the user program.

If an excess scan time is identified, the status of the PLC will be changed to ”User Program HALT” (the networking to the PADT will be interrupted for a few seconds). After this it is only possible to restart via the PaDT (power–up with B1 set to ”Automatic Start” no longer causes the user program to start automatically).

”Watchdog Time” determines the maximum scan times for the OB1, interrupt–driven OBs and time–controlled OBs running in timesharing-mode. Time–controlled OBs not running in timesharing mode will automatically have a timeout time of 2 * time base.

The watchdog time for the ALU 204/205 is specified by power supplies DNP 205 and DNP 210 as 300 ms. In order that time for the ALU 204/205 can be managed just like time for the ALU types ALU 151 ... 154, watchdog time is retriggered by the software if a time greater than 300 ms is set.

The values that can be specified depend on the ALU type and the SW watchdog:

- SW Watchdog ”No”
  - ALU 151 ... 154: 100 ms ... 1 600 ms
  - ALU 204/205: 300 ms ... 1 600 ms

- SW Watchdog ”Yes”
  - ALU 151 ... 154: 300 ms ... 1 600 ms
  - ALU 204/205: 300 ms ... 1 600 ms

Equipment List Editor / Early–Warning time watchdog

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>,
  ”Parametrize PLC”, ”Early–Warning time watchdog”

If the ”Early–Warning Time Watchdog” is exceeded, system marker SM164 is set; the user program keeps running. This monitoring is only active for the OB1.

Values from 100 ms to 1 500 ms are permissible.

Equipment List Editor / Scan time I/O Error handling

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>,
  ”Parametrize PLC”, ”Scan time I/O Error handling”

I/O nodes identified as faulty will be automatically reinterfaced once the error has been eliminated. The time base for this operation can be set anywhere between 30 and 10 000 ms. This setting should not be smaller than necessary, in order to keep the base load minimal.

Equipment List Editor / Insert Line (Move Node)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>,
  ”Move Node”

The module highlighted by the cursor (and all those following) is moved back one line to make a gap where a new module can be entered.

Equipment List Editor / Erase Line (+ Remove Node)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>,
  ”Close Gap”

The module or blank line highlighted by the cursor is removed; following nodes move up one position.

Note Node numbers (TNs) are **not** changed by this action.

Index

A

A120, Programming, 3
-A250 subrack, Start Entry, 197
-A250 subrack, start entry, 201
-A250 expansion
  -DEA interface address, 197, 198, 201, 202
  Initial node address, 198, 201
  Networking structure, 198
Address input
  FBD, 88
  LD, 66
Addressing, 60–61
  Pointers, 129
ALU 204, 3
ALU 205, 3
-ALU select.
  -ALU type, 179
  Enter, 178
  -Floating point, 179
-ALU type, 179
Analog I/O, 183
-AND block, 93
Area
  Copy, 162
  Erase, 163
  Move, 164
-Arithmetic functions, IL, 134
--Auto Repeat Function, 22

B

-Equipment List, InterBus, 206
-Equipment List incompatibility, 172
Block, Edit, 100
Block call
  conditional, 101, 102
  unconditional, 101, 102
-Block Editor, 34
Break
  Correction-mode, 45
  Edit, 55
  Equipment list, 195
Bus terminal, 204
-Bus terminal
  -Inst. remote bus, -Slot reservation, 205
  -Inst. bus, -Slot allocation, 208
  Inst.-bus, -Slot allocation, 207

C

Comment line, Insert, 169
Comments, for signals, 158
Comments-editor, IL, 47
Comparator
  IL, 137
  LD/FBD, 152
Connector, 74, 82, 98
Continuation of parallel path, 83
Controller, Parameterize, 213
Copy
  Area, 162
  Modified, 165
  Network, 40
Copy Comments, 48
-Correction mode, 67, 88
Correction-mode, 41
Counter down CD, IL, 115
Counter down CD, LD/FBD, 147
Counter range CU, CD, CUD, 115, 116

Counter up CU, IL, 115
Counter up CU, LD/FBD, 146
Counter up down CUD, LD/FBD, 148
Counters, 218
  IL, 115
  LD/FBD, 146
Cross–connection of parallel path, 83
Cursor Positioning, 51
Cyclic action block AZ, 142, 144
Cyclic processing of modules, 220
Cyclic scheduling, 188

D

Data blocks, 218
Data structure for SFBs, 108
Data type, 190
Databank, 154
DEAKOP, Subrack networking, 198
Decrementing, DEC, 119
Delete, Network, 39
Destination area, 166
Differences between versions, 5
Digital I/O, 181
Direct scheduling, 188
Display Formats IL, LD, FBD. *Siehe* Eingabe-Modus
does not equal, IL, 137
does not equal, LD/FBD, 152
Dynamic Status Display, Network, 58

E

Early–Warning time watchdog, 228
Edge recognition, LD/FBD, 153
Edit, 10, 13, 26, 29
  Block, 31, 100
  Blocks, 10, 13, 33–99
    OB, PB, FB, AE, AZ, 34
  Compile Blocks, 10, 13
  Data structures, 10
  Equipment List, 10
  Equipment list, 13, 172–228
  Network number, 108
  Overview, 10, 13, 30–32
  Read Out PLC, 10
  Read–out PLC, 13
  Replacement of Signals, 10, 13
  -Run–time system, 10
  Symbols and Comments, 10, 13, 154–171
  Title Block, 10, 13
  Compile data -blocks, 13
Editing in various display formats. *Siehe* AWL-Editor, FUP-Editor, KOP-Editor
Eight–word block, 61
Element, 221
Enter, 224
  Comments, 213
Enter Symbol in SYM/COM block, 52
Enter Timeout, 213
equal to, IL, 137
equal to, LD/FBD, 152
Equipment list, Standardize, 223
Equipment list column 1, 178
Equipment list column 2, 178–187
Equipment list Column 3, 187
Equipment list Column 4, 188
Equipment list column 5, 190
Equipment list column 6, 190
-Equipment List Editor, 194
Erase
  Area, 163
  Field, 161
  Line, 44, 161
  Module, 193
  Network, 79, 99
  Subrack, 210
Erase Module, -Inst. remote bus, 204
Exchange, Online, 57
--Exclusive OR block, 76, 97
Experts. *Siehe* intelligente Funktionsbaugruppen

F

-FBD Editor, 53, 54, 55, 56, 57, 58, 60, 77, 78, 79, 84, 88, 91, 92, 93, 94, 95, 96, 97, 98, 99, 145, 153
-FBD editor, 153

-FBD Elements, 79, 143, 145
Erase, 65
Insert, 65
-FBD elements, 143
FBD-Editor, 39, 40, 41, 45, 46, 48, 50, 51, 52
FBD-Elements, -Correction mode, 48
Field, Erase, 161
FL, 118
LD/FBD, 153
FLN, 118
LD/FBD, 153
-Floating point, 179
FLP, 118
LD/FBD, 153
Function block diagram FBD, 101, 144, 145, 146, 150, 151, 152, 153
Function block FB, 100, 102, 142, 144
Call, 107
Declaration part, 102
Edit, 103
Instruction part, 102, 106
Function keys, 17

G

Gaps, Slot, 197, 200
Generate Symbols, 169
greater than, IL, 137
greater than, LD/FBD, 152
greater than/equal to, IL, 137
greater than/equal to, LD/FBD, 152

H

Hard copy. *Siehe* Bildschirm-Kopie
Help text, 28

I

IB, 158
-IL Editor, 145
IL-Editor, 36, 38, 39, 40, 41–42, 43, 44, 45, 46, 48, 49, 50, 52, 53, 54, 55, 56, 57, 58, 60
Correction mode, 47
Incrementing, INC, 119
Indirect addressing, 127
Initial values, 154
Initial values for signals, 158
Incompatibility of the Equipment List-, 172
Input, 95
Input mode, 60
Display Formats IL, LD, FBD, 108
Input Symbols and Comments, 156
Insert
Comment line, 169
Line, 43
Example, 43
Network, 39
-Inst. remote bus, Erase Module, 204
Installation remote bus, -Slot allocation, 207, 208
-Installation remote bus, -Slot reservation, 205
Instruction list IL, 101, 109, 110, 112, 114, 115, 117, 118, 119, 120, 121, 127, 132, 133, 134, 136, 137, 138, 140, 141
Intelligent modules (Experts), 184
InterBus, Equipment list, 177
-InterBus module types
Auswahl, 204
DTA 202 on the remote bus, 204
Inst.-bus-definition, 204
Selection, 202, 203
-InterBus, -Subrack expansion, 200

J

Job elements, 220
Jump, conditional SPB, IL, 132
Jump, unconditional SP, IL, 132

K

Key assignment
+, 55
–, 55
Alt+F2, 20
Alt+F8, 21
Alt+F9, 22

Alt+letter, 21, 22
Alt+number, 18
Arrow keys, 17, 19, 23, 24, 28, 31, 38, 42, 47, 64, 69, 86, 90, 159, 177, 220
backspace, 19, 159
backtab, 42, 47, 159
Bild, 17
Character, 90
Characters, 30, 69
Ctrl, 17
Ctrl+A, 19
Ctrl+Arrow keys, 69, 159
Ctrl+B, 220
Ctrl+C, 24
Ctrl+D, 19
Ctrl+End, 177
Ctrl+F, 19
Ctrl+Home, 177
Ctrl+I, 69, 90
Ctrl+M, 19
Ctrl+Pfeiltasten, 90
Ctrl+PgDn, 42, 69, 90, 159, 177
Ctrl+PgUp, 42, 69, 90, 159, 177
Ctrl+R, 19
Ctrl+Reference characters, 23
Ctrl+Return, 64, 69, 86
Ctrl+Right arrow, 90
Ctrl+S, 19
Ctrl+Y, 19
Del, 17, 19, 38, 39, 42, 47, 64, 69, 86, 90, 159
Druck, 17
Einfg, 17
Eing Lösch, 17
End, 17, 19, 38, 47, 64, 86, 90, 159, 177
Ende, 17
Esc, 17, 19, 24, 31, 32, 38, 42, 47, 59, 64, 69, 86, 90, 159, 177, 220
F1, 69, 90
F10, 2, 28
F2, 69, 90
F3, 69, 90
F4, 69, 90
F5, 90
F6, 69, 90
F7, 69
F9, 69, 90
Fx, 159
Graphics characters, 18
Home, 17, 19, 38, 41, 47, 64, 86, 159, 177
Ins, 17, 19, 24, 38, 39, 42, 47, 64, 69, 86, 90, 159
Lösch or Entf, 17
PgDn, 17, 31, 38, 42, 64, 69, 86, 90, 159, 177
PgUp, 17, 31, 38, 42, 64, 69, 86, 90, 159, 177
Pos1, 17
PrtSc, 17
Reference characters, 17
Return, 19, 23, 24, 28, 30, 47, 69, 90, 159, 220
shift+tab, 177
Strg, 17
tab, 42, 159, 177
-Key input memory. *Siehe* Lernen
-Key macros, 19, 22

L

Ladder diagram LD, 101, 142, 143
-LD Editor, 145
LD-Editor, 39, 40, 41, 45, 46, 48, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 62, 64, 67, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 86
LD-Editor, Correction mode, 69, 90
LD-Elements, Erase, 65
LD-elements, Insert, 65
Learning, 21
less than, IL, 137
less than, LD/FBD, 152
less than/equal to, IL, 137
less than/equal to, LD/FBD, 152
Line
Erase, 44, 161
Insert, 43

Example, 43
Line comment, IL, 38
Line editor, 19
Load, 10, 13, 26
Bootload, 10, 13
Compare, 10, 13
Edit EPROM, 10
Exchange Online, 10, 13
-Expert Data, 10
Link Program, 10, 13
Process EPROM, 13
-Process PLC data, 10
Program to PLC, 10, 13
Read Equipment List, 10, 13
Set date/time, 10, 13
Load High Data, LHD, 125
Load Low Data, LLD, 125
-Logic functions, IL, 112–127

M

Marker Bits, 216
Marker Bytes, 216
Marker Counters, 218
Marker Double Words, 217
Marker Floating Point Words, 217
Marker Times, 217
Marker Words, 216
Max. Number of tasks, 225
Memory
IL, 114
LD/FBD, 145
Modified, Copy, 165
Modify, Network, 41
Modify BIK / DEA, 211
Module, Erase, 193
Modules
Analog I/O, 183
Digital I/O, 181
Experts, 184
Special (Power supplies etc.), 186
Mouse, 16
Move, Area, 164
-Operating sys. MS–DOS. *Siehe* Betriebssystem
MS-DOS. *Siehe* Betriebssystem
Multiple output assignment, 75

N

-Near–data structures, 218
Negated input, 94
Nesting function blocks, 107
Network
Copy, 40
Delete, 39
Erase, 79, 99
Insert, 39
Modify, 41
Search, 56
Network number, Edit, 108
Network-comments, 46–47
Network-header, 46
-Node number, 190
Direct scheduling, 188
SFB (TN), 185
Normally closed contact, 73, 82
Parallel path, 73
Serial path, 73
Normally open contact, 72, 81
Null pointer, 131
Number of tasks, 225
Numerical pad, 18
NW End Character, 49

O

Off–delay TA
IL, 117
LD/FBD, 151
On–delay TE
IL, 117
LD/FBD, 150
On–delay, stored TS, IL, 117
Once–only action block AE, 142, 144
Online, 11, 13, 26
Communication Operating Functions, 11
Dynamic state display, 13
Dynamic Status Display, 11

-Online–List, 11, 13
-PLC status, 11, 13
-Run–time system, 11
-PLC diagnostics, 11
Start PLC, 11, 13
Stop PLC, 11, 13
Operand, Search, 161
-OR block, 96
Organization block OB, 100–101, 142, 144
Output, 75
Output monitoring, 51
Output, LD-correction
Existing network, 75
New network, 75

P

PaDT, 18, 22
Parallel path, 80
Parameter, -Equipment List, 175
Parameterize, Central Controller, 213
-Parameterize PLC, 224
Parameterize signal memory. *Siehe* Zentrale parametrieren
Path, explanation, 26
PB-/FB-/SFB-Conditional call, 77
PB-/FB-/SFB-Unconditional call, 78
Permissible operations, Pointers, 130
Phase, 221
Pointer option, 130
Pointers, 218
Addressing, 129
Permissible operations, 130
Presetting
Correction-mode, 50
Network, 60
Primary backplane, Modify, 223
Print, 11, 14, 26
Complete documentation, 11, 14
Cross–reference list, 11, 14
Data structures, 11
Equipment list, 11, 14
Overview, 11, 14
PLC-data, 11
Program -protocol, 11, 14
Signal-assignment list, 11, 14
Symbols and Comments, 11, 14
Program block PB, 100, 101, 142, 144
-Program structure, 30
Programming, A120, 3
-Pulldown menu
-ALU select., 178
Edit Blocks, 33, 35
Edit Network, 37
Edit network, 63
Module type, 180
Program structure, 30
Pulldown- menu, IL- Editor, 37
-Pulldown menus
Analog I/O, 183
Choice of Subracks, 197, 200
Choice of subracks-, 200
Copy Area, 162
Digital I/O, 181
Dyn. Status Display, 59
Edit, 29
Edit FBD, 89
Edit LD, 68
Edit network, 85
Edit Symbols and Comments, 160
-EQL Editor Menu, InterBus, 194
-EQL Editor menu, 194
-Equipment List Editor, InterBus, 194
Erase Area, 163
Experts, 184
-FBD Elements, 89
Job Elements, 220
Linker, 199
Main menu, 27
Move Area, 164
Parameterize Central Controller, 214
Presetting, 60
Replace Text, 168
Search Function, 160
Search Text, 167
-Special modules, 186

Subrack, 177, 200, 223
-SYM/COM – Editor, 160
Pulldown-menu, Edit Network, 35
Pulldown-menus
Copy Network, 40
Edit comments, 46
Edit IL, 42
-Equipment List Editor, 194
FBD-Elements, 48
Select Editor Setting, 50
Pulse TI
IL, 117
LD/FBD, 150
Pulse, extended TV
IL, 117
LD/FBD, 150

R

Reference characters, 17, 23, 24
reference characters, 26, 28
Remote bus, Bus terminal, 204
Replace, Text, 168
Reserve, 218
Rotate left, ROL, 119
Rotate right, ROR, 119
RS memory, LD/FBD, 145
RS-memory, IL, 114
Rules for instruction lists
Angle functions, 141
Bracket operations, 111
Calculating, 135
Comparator, 137
Connector, 113
Constructing an IL, 110
Conversion, 138
Counters, 115
Edge recognition, 118
Exponential functions/Sign handling, 140
-FREE instruction, 133
Incrementing/decrementing, 119
Indirect addressing, 127
Initializing operations, 112
Initiating pulse, 110
Jumps, 132
Load
Arithmetic, 134
-Operand address, 134
Load, Transfer, 120, 121
Logic, 120
Marker bit, 110
Modulo function, 141
-NOP instruction, 133
Order of logic operations, 112
Pointers, 127
Rotate, 119
Save operations, 114
Shift, 118
Timers, 117
Two’s complement, 136

S

Scan, 221
-Scan time I/O Error handling, 228
-Screen copy, 20
Screensave. *Siehe* Bildschirm-Kopie
Scroll Backward, Network, 54
Scroll backward, Overview, 31
Scroll Forward, Network, 53
Scroll forward, Overview, 31
Search
Block, 32
Network, 56
Operand, 161
Text, 167
Search Function, 160, 195
Setpoint value, 148
SeTup, 14, 26
Colors, 14
Plant, 14
-PLC station, 14
Print, 14
Setup, 12
Colors, 12
Networking, 12
-PADT station, 12
-PLC station, 12
Print, 12

System, 12
Shift left, SHL, 118
Shift right, SHR, 118
Signal, Search, 55
Slot gaps, 197, 200
Slot-independent. *Siehe* Teilnehmeradressen-unabhängig
Slot-independent programming, 192
-Software, compatibility, 9
Special, 12, 14, 26
Backup the station, 12, 14
Copy diskettes, 12, 14
Copy files, 12, 14
Duplicate station, 12
End station handling, 12, 14
Erase a Station, 12, 14
Erase Files, 12, 14
Export, 12
Format diskettes, 12, 14
-Further process PLC data, 12
Import, 12
Operating sys. OS/2, 12, 14
Restore a station, 12, 14
System information, 12, 14
Table of contents, 12, 14
Special keys, 23
-Special modules, 186
Spread Horizontal, 70, 91
Spread Vertical, 71, 92
-PLC memory test, 225
-SR memory, IL, 114
SR memory, LD/FBD, 145
-Standard function block SFB, 108
TN, 185
Standard-Funktionsbaustein SFB, TN, 191
Standardize, Equipment List, 223
Start Entry, -A250 subrack, 197, 201
Stored on–delay TS, LD/FBD, 151
Subrack, Erase, 210
Subrack size, InterBus, 194
Subrack, Enter, 197
-SW time Monitoring, 226
-SYM/COM block, 154
-SYM/COM Editor, 155
Symbols for signals, 157
System limits, 5
System markers, 215

T

Task data memory, 224
Tastentabelle, F5, 69
Teilnehmer-Nummer, SFB (TN), 191
Terminate
Correction-mode, 45
Edit, 55
Equipment list, 195
Overview, 32
Text
Replace, 168
Search, 167
Time base TI, TV, TE, TS, TA, 117
Timers, 217
IL, 117
LD/FBD, 150
-TN No., 190
Direct scheduling, 188
TN, -Node number SFBs, 185
TN, Teilnehmer-Nummer SFBs, 191
Toggle, 24
Transition block TB, 142, 144
-Typewriter keyboard, 17, 23

U

Unlinked blocks, 32
-Using the keyboard, 17–24

V

Variants, 187

W

Watchdog time, 227